Homework 1

Welcome to the first homework assignment!

The purpose of this homework is to practice using R and R Markdown, and to review some concepts from introductory Statistics. Please fill in the appropriate R and R Markdown and write answers to all questions in the answer section., then submit a compiled pdf or html with your answers to Canvas by 11:59pm on Sunday September 8th.

If you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Problem 1: RMarkdown practice

RMarkdown has a number of features that allow the text in your written reports to have better formatting. In the following exercise, please modify lines of text to change their formatting. A cheatsheet for RMarkdown formatting can be found here. When answering the questions (i.e., formatting the text below) be sure to knit your RMarkdown document very often to catch errors as soon as they are made.

Problem 1.1: Please format the lines of text below (15 points)

Make this line bold

Make this line italics

Make this line a third level header

- Make this line a bullet point

LINK (make the word LINK at left link to Yale's home page)

Problem 1.2: Use LaTeX to write plato's name in Greek below (10 points)

Note: make sure the ending dollar sign touches the last letter otherwise you will get an error when knitting.

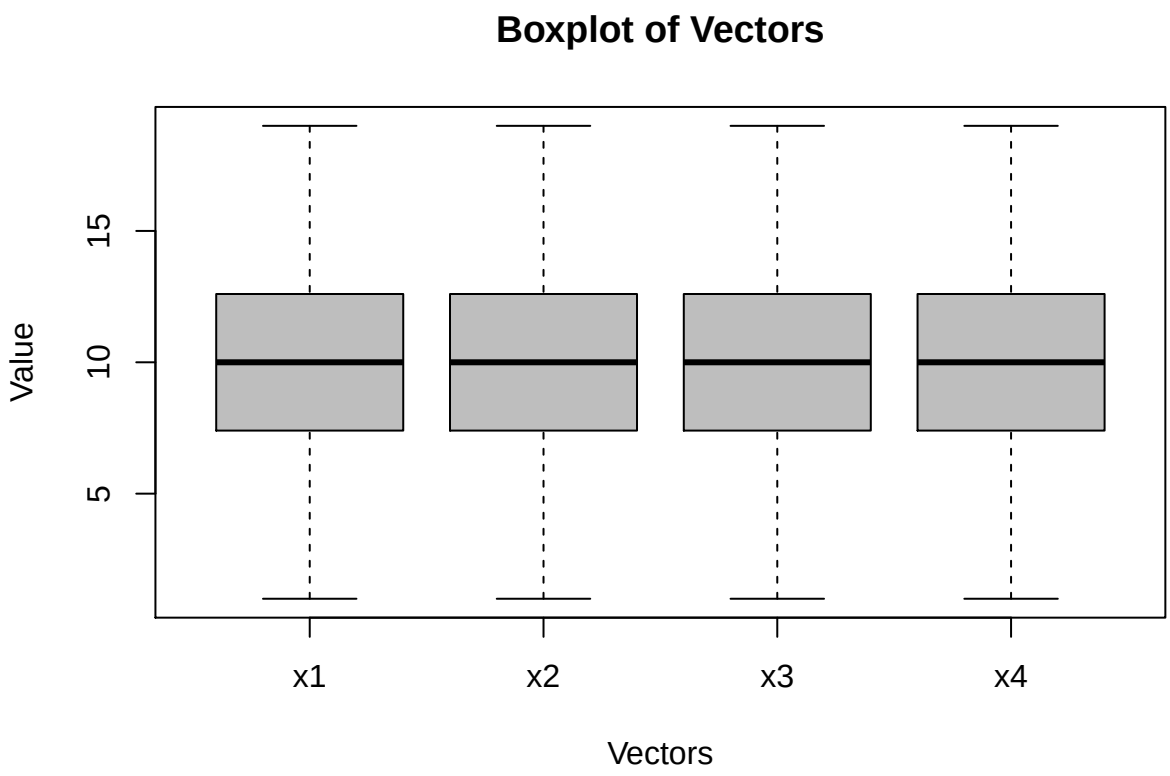
$\Pi\lambda\acute{\alpha}\tau\omega\upsilon$

Problem 2: Descriptive statistics and plots

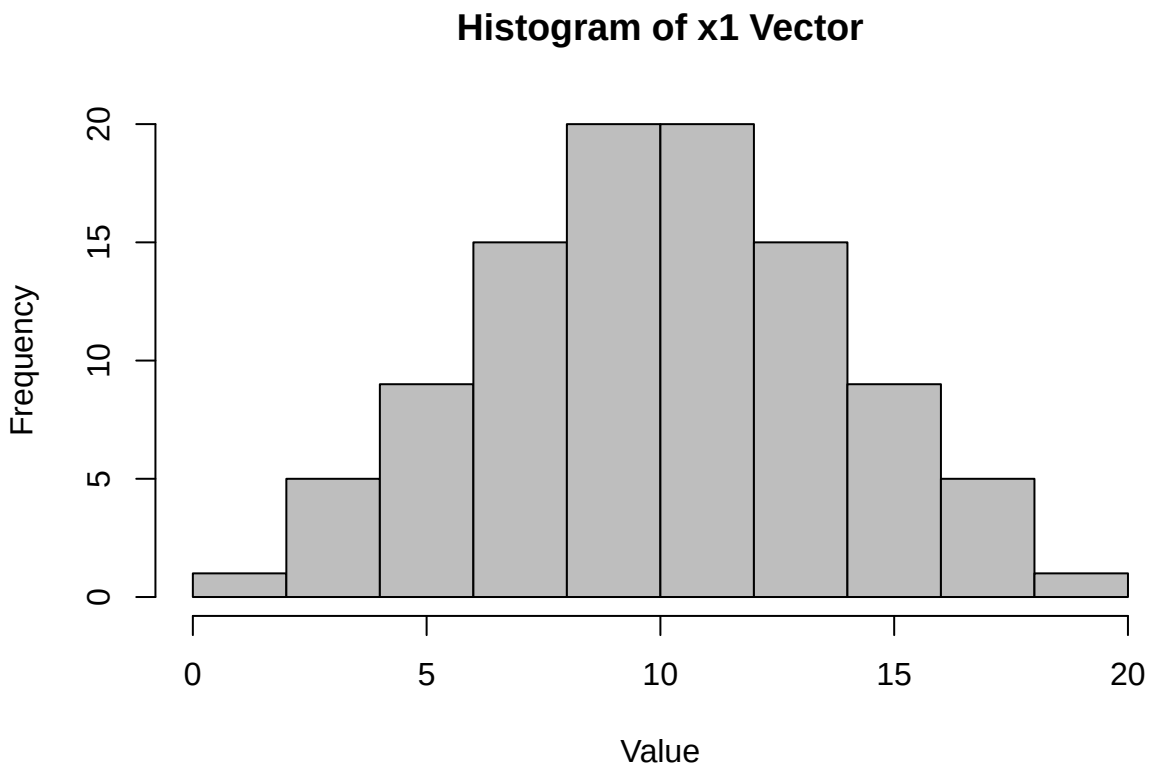
Below you will create and compare a few plots. Please answer each question, and if you notice any outliers in your data please address them appropriately. Also be sure to label your plots appropriately.

Part 2.1: (10 points) The code chunk below loads four vectors objects named x1, x2, x3, and x4. Create a side-by-side boxplot that compares these four vectors. Also create a histogram for each of these vectors (4 histograms total). Describe below whether the boxplots or histograms are more informative for plotting this data and why.

```
load("misc_data.Rda")
boxplot(x1, x2, x3, x4, names = c("x1","x2", "x3", "x4"), xlab = "Vectors",
        ylab = "Value", main = "Boxplot of Vectors", col = "gray")
```

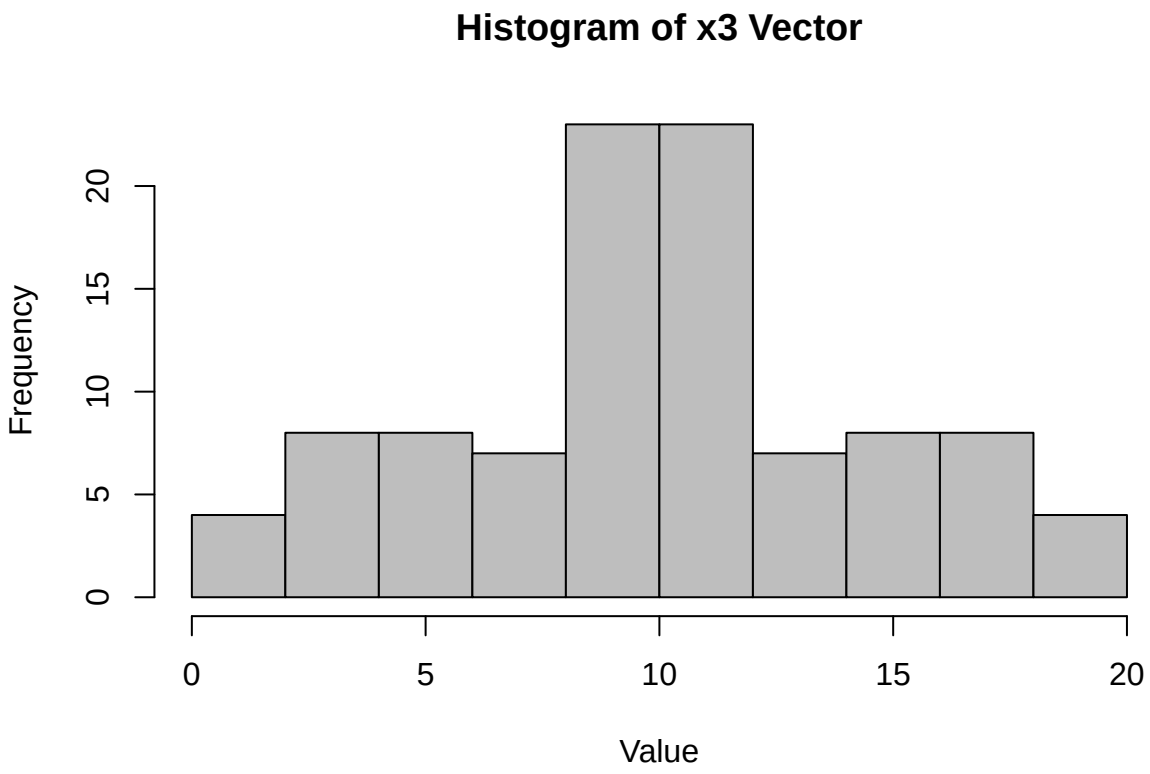


```r
hist(x1, xlab = "Value", ylab = "Frequency", main = "Histogram of x1 Vector", col = "gray")
```




```r
hist(x2, xlab = "Value", ylab = "Frequency", main = "Histogram of x2 Vector", col = "gray")
```


```
hist(x3, xlab = "Value", ylab = "Frequency", main = "Histogram of x3 Vector", col = "gray")
```




```
hist(x4, xlab = "Value", ylab = "Frequency", main = "Histogram of x4 Vector", col = "gray")
```

Answer: [Describe whether boxplots or histograms are more informative here]

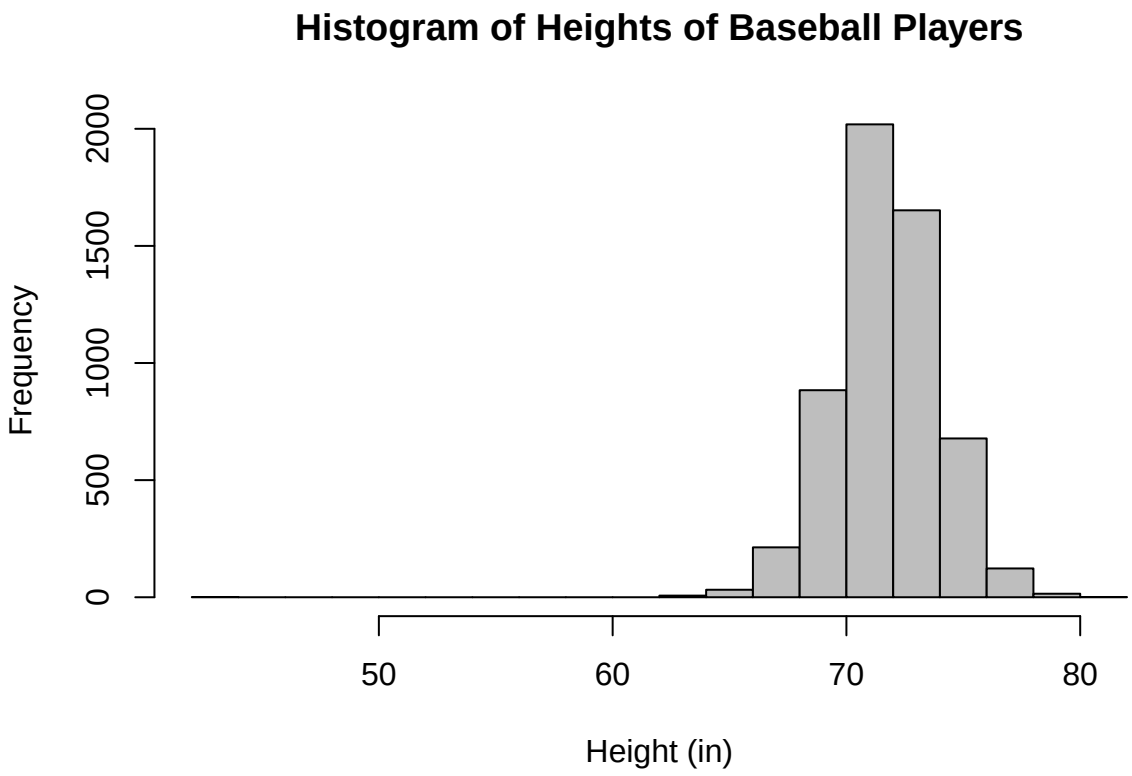
The histograms are more informative here. With the boxplot, we can only see summary statistics such as the median, quartiles, and extremes. In regards to this data set, for x1, x2, x3, and x4, all of these values are exactly the same. However, with the histograms, we can see the actual distribution of data points. The histograms reveal that the data sets are actually quite different despite having the same summary statistics.

Part 2.2: (10 points) The R chunk below loads a data frame with information on all major league baseball players who were born between 1901 and 1950 (if you are interested in the data, it comes from the Lahman package). Create a vector object that is called `heights`, that has just the player heights. Then create a histogram and a boxplot of the players' heights using this vector object. Describe the shape of the distribution of heights and any advantages that one type of plot has over the other. Also investigate any unusual features of the data.

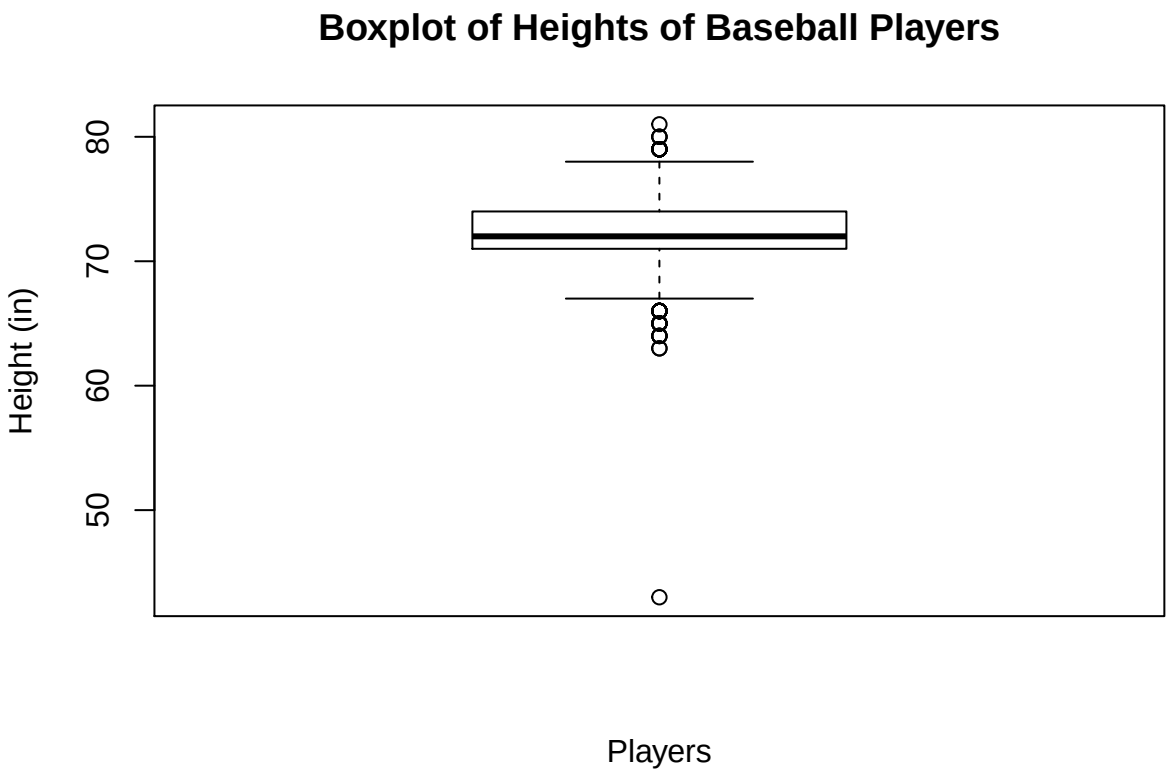
```
load("players_born_1901_1950.Rda")
heights <- players_born_1901_1950$height
mean(heights, na.rm = TRUE)
```

```
## [1] 72.23307
```

```
hist(heights, main = "Histogram of Heights of Baseball Players", xlab = "Height (in)",
     ylab = "Frequency", col = "gray")
```




```
boxplot(heights, main = "Boxplot of Heights of Baseball Players", ylab = "Height (in)",
      xlab = "Players")
```

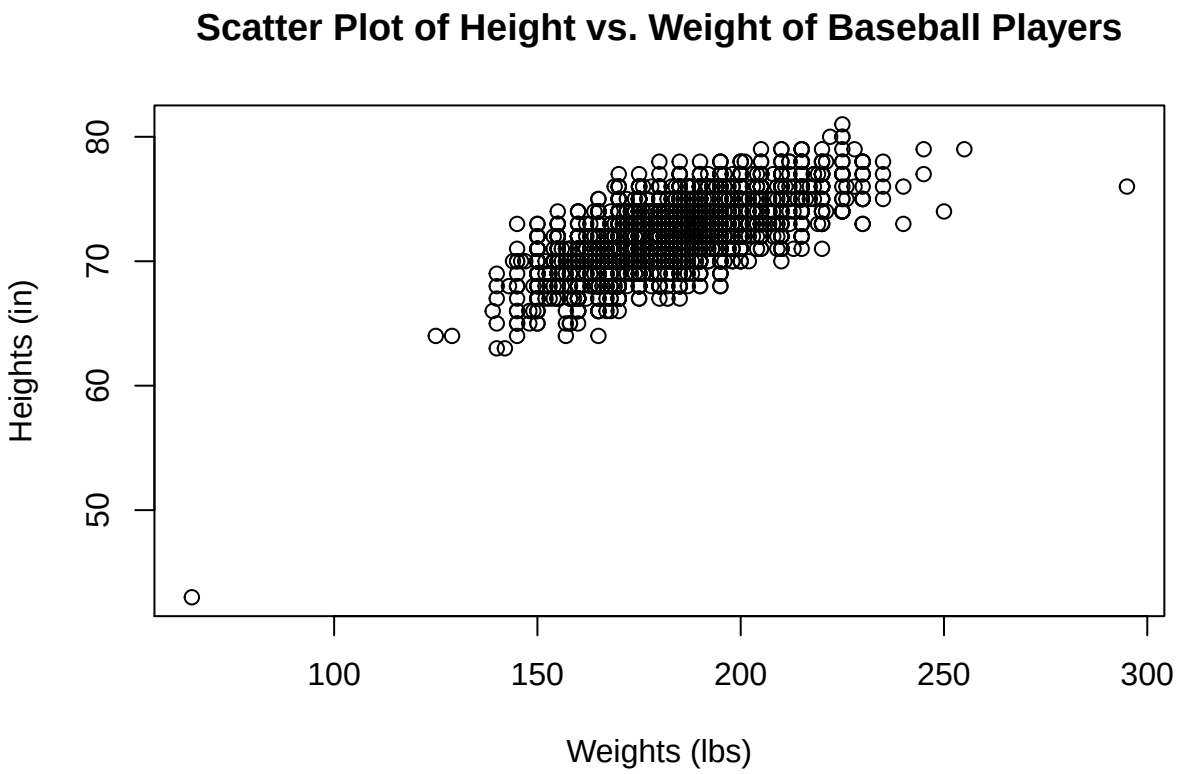


Answer: [Describe advantages of boxplots and histograms for this data and investigate usual features of the data]

The boxplot here tells us that the heights of baseball players, with the exception of one outlier, lie mostly close to the median height. The boxplot also allows us to see the summary statistics of the data, such as the quartiles, whereas the histogram allows us to see the distribution of the data (we can see that the heights are approximately normally distributed about a mean height of 72.23 inches, calculated above). The histogram also excludes the lowest height value, an outlier of 43 inches, as a result of its scaling, so the advantage of the boxplot is that the outlier is shown more clearly.

Part 2.3: (10 points) Create a scatter plot of the baseball player's heights as a function of their weight. Describe what the results show.

```
weights <- players_born_1901_1950$weight
plot(weights, heights, main = "Scatter Plot of Height vs. Weight of Baseball Players",
     xlab = "Weights (lbs)", ylab = "Heights (in)")
```



Answer:

The height vs. weight scatterplot shows a positive correlation between height and weight. This makes sense since taller people tend to weigh more.

Problem 3: Examining categorical data

Let's now examine which states/regions baseball players are born in.

Part 3.1: (10 points) Use the table() function to create an object called birth_place_counts that has the counts of where players were born in. What is the state that the most players were born in?

Then create a bar plot and pie chart showing the counts of places that players are born in. How do these plots look? How could we make them better?

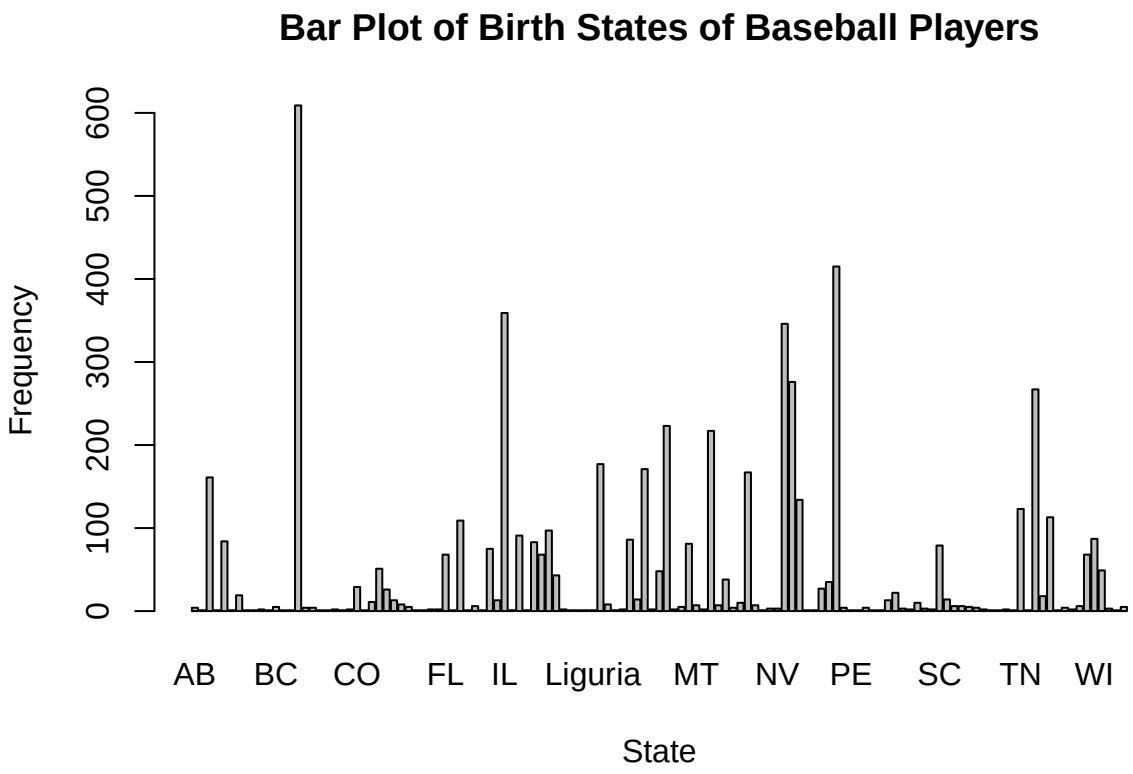
```
birth_states <- players_born_1901_1950$birthState
birth_place_counts <- table(birth_states)
sort(birth_place_counts, decreasing = TRUE)
```

```
## birth_states
##                  CA                  PA                  IL
```

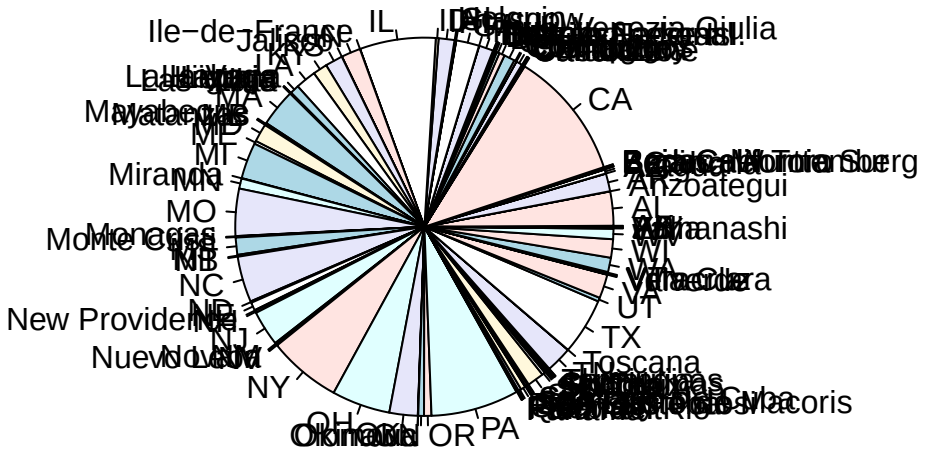
```
##                 609                 415                 359
##                  NY                  OH                  TX
##                 346                 276                 267
##                  MO                  NC                  MA
##                 223                 217                 177
##                  MI                  NJ                  AL
##                 171                 167                 161
##                  OK                  TN                  VA
##                 134                 123                 113
##                  GA                  LA                  IN
##                 109                  97                  91
##                  WI                  MD                  AR
##                  87                  86                  84
##                  KS                  MS                  SC
##                  83                  81                  79
##                  IA                  FL                  KY
##                  75                  68                  68
##                  WA                  CT                  WV
##                  68                  51                  49
##                  MN           La Habana                  NE
##                  48                  43                  38
##                  OR                  CO                  ON
##                  35                  29                  27
##                  DC                  RI                  AZ
##                  26                  22                  19
##                  UT                  ME                  SD
##                  18                  14                  14
##                  DE                  ID                  QC
##                  13                  13                  13
##               Colon                  NH  San Pedro de Macoris
##                  11                  10                  10
##   Distrito Federal            Matanzas                  MT
##                   8                   8                   7
##                  ND                  NM                  HI
##                   7                   7                   6
##             Sinaloa                  SK                  VT
##                   6                   6                   6
##                  BC   Distrito Nacional        Monte Cristi
##                   5                   5                   5
##              Sonora               Zulia                  AB
##                   5                   5                   4
##            Camaguey         Canal Zone      New Providence
##                   4                   4                   4
##              Panama       Pinar del Rio           St. Croix
##                   4                   4                   4
##            Veracruz         Nuevo Leon                  NV
##                   4                   3                   3
##       San Cristobal            Santiago                  WY
##                   3                   3                   3
##  Baja California Sur           Chihuahua          Cienfuegos
##                   2                   2                   2
##            El Seibo              Falcon            La Vega
##                   2                   2                   2
##                  MB             Miranda             Monagas
```

```
##                   2                   2                   2
##                  NB      San Luis Potosi      Santiago de Cuba
##                   2                   2                   2
##          St. Thomas           Tamaulipas           Villa Clara
##                   2                   2                   2
##                  AK           Anzoategui               Aragua
##                   1                   1                   1
##    Baden-Wurttemberg      Baja California             Barahona
##                   1                   1                   1
##              Berlin       Bocas del Toro             Carabobo
##                   1                   1                   1
##            Cheshire             Chiriqui             Coahuila
##                   1                   1                   1
##     Dodescanese Isl.              Duarte Friuli-Venezia Giulia
##                   1                   1                   1
##              Glasgow             Holguin        Ile-de-France
##                   1                   1                   1
##              Jalisco                Lara           Las Villas
##                   1                   1                   1
##              Liepaja             Liguria            Mayabeque
##                   1                   1                   1
##               Novara             Okinawa             Olomouc
##                   1                   1                   1
##                  PE            Piedmont            Plzensky
##                   1                   1                   1
##              Puebla               Sucre             Suffolk
##                   1                   1                   1
##            Thuringia             Toscana             Valverde
##                   1                   1                   1
##            Yamanashi
##                   1
```

```
barplot(birth_place_counts, main = "Bar Plot of Birth States of Baseball Players",
        xlab = "State", ylab = "Frequency")
```




```
pie(birth_place_counts, main = "Pie Chart of Birth States of Baseball Players")
```



Answers: The most players are born in California (we can see this when we sort the table, or after we create the bar and pie charts).

Both the bar plot and pie chart are oversaturated with data points, making the labels impossible to read (in the case of the pie chart) or missing (in the case of the bar plot, where there is not enough space for the labels). The plots should be reformatted to include only essential information (for example, taking only points with large enough values).

Part 3.2: (10 points)

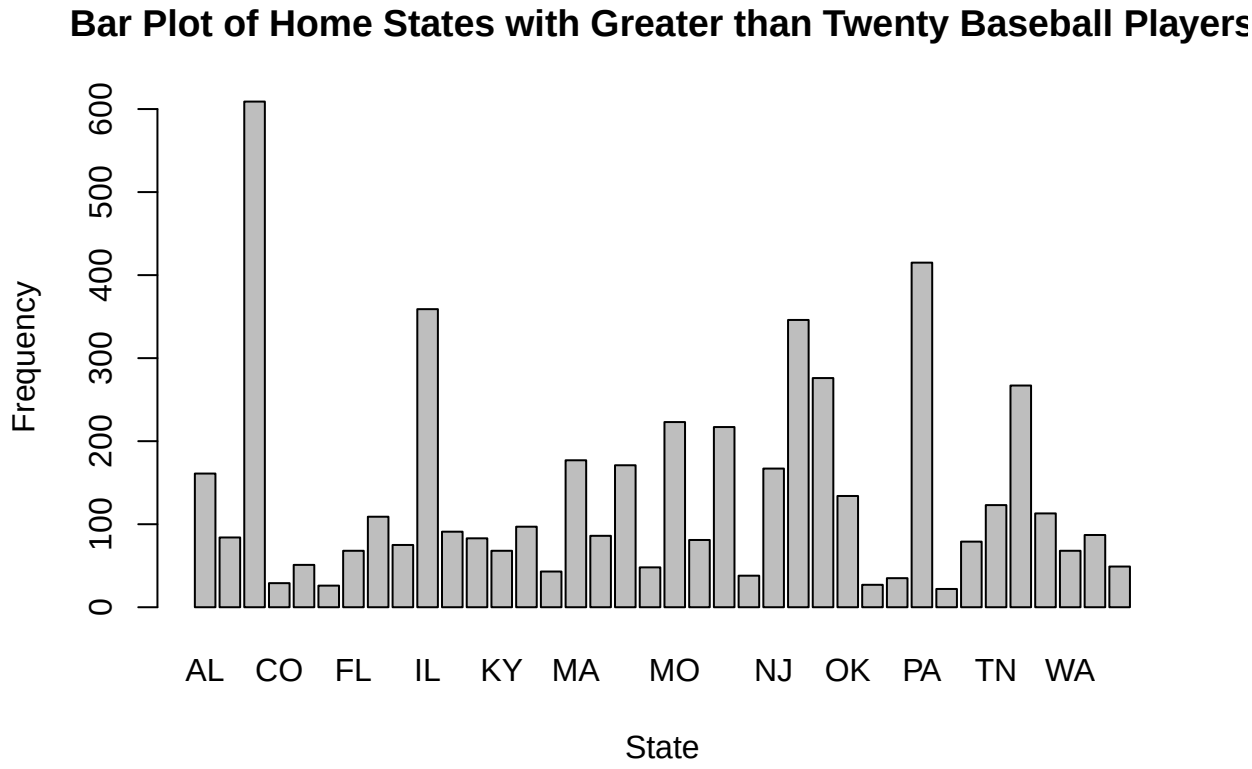
Let's only plot states/places that have more than 20 players born in them. You can do this by creating a vector of booleans where TRUE indicates a state that has greater than 20 players born in it and FALSE indicates that 20 or less players were born in it (this can be done in 1 line of code). Then use this vector to extract only the places which more than 20 players born in. Finally replot the results with only states with more than 20 players born in them. Does this look better? Is there any place on this list that is not a state?

```
twenty_players_vec <- (birth_place_counts > 20)
twenty_players <- birth_place_counts[twenty_players_vec]
twenty_players
```

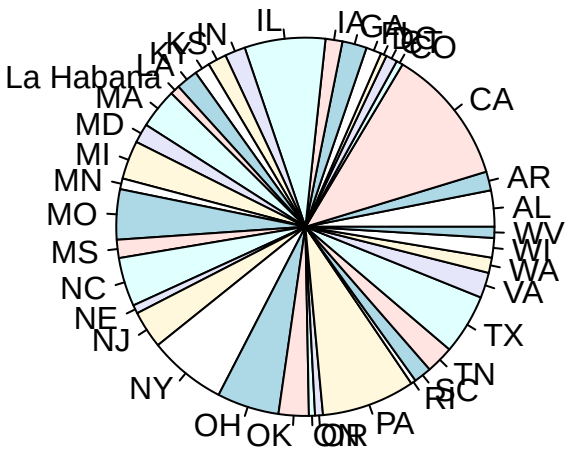
```
## birth_states
##        AL        AR        CA        CO        CT        DC        FL
##       161        84       609        29        51        26        68
##        GA        IA        IL        IN        KS        KY        LA
##       109        75       359        91        83        68        97
```

```
## La Habana         MA         MD         MI         MN         MO         MS
##         43        177         86        171         48        223         81
##         NC         NE         NJ         NY         OH         OK         ON
##        217         38        167        346        276        134         27
##         OR         PA         RI         SC         TN         TX         VA
##         35        415         22         79        123        267        113
##         WA         WI         WV
##         68         87         49
```

```
barplot(twenty_players,
        main = "Bar Plot of Home States with Greater than Twenty Baseball Players",
        xlab = "State", ylab = "Frequency")
```




```
pie(twenty_players,
    main = "Pie Chart of Home States with Greater than Twenty Baseball Players")
```



Answer: The data is clearer and easier to read, although some labels are still missing on the bar plot due to having too many bars, and labels on the pie chart are still difficult to read due to overlapping. The vector includes data for La Habana, a city in Cuba (not a state).

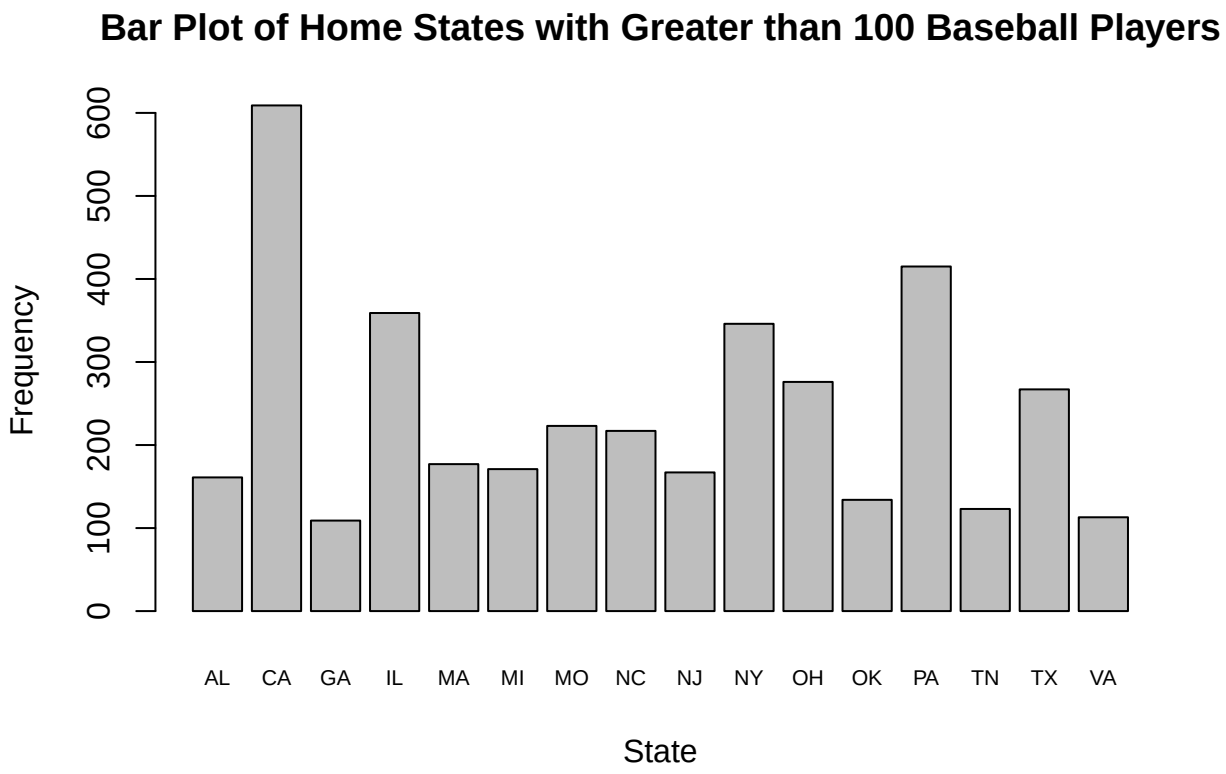
Part 3.3: (10 points)

The plots in part 3.2 still could look better. Adjust the plots so that you plot fewer states so that it is easier to see exactly which states most players are born in. Also adjust other visual attributes of the plots so that none of the labels are overlapping, and see if you can find other ways to make the plots look better, e.g., by adjusting the colors, etc. (hint: using ? pie and google will be helpful). Is plotting only some of the states misleading in any way, and if so, what are ways this could be addressed?

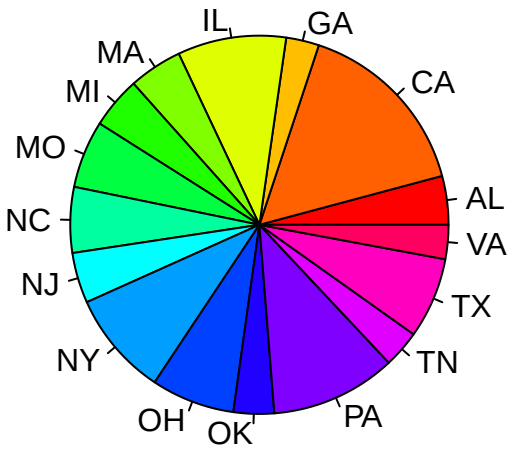
```
state_players_vec <- (birth_place_counts > 100)
state_players <- birth_place_counts[state_players_vec]
state_players
```

```
## birth_states
##  AL  CA  GA  IL  MA  MI  MO  NC  NJ  NY  OH  OK  PA  TN  TX  VA
## 161 609 109 359 177 171 223 217 167 346 276 134 415 123 267 113
```

```
barplot(state_players, main = "Bar Plot of Home States with Greater than 100 Baseball Players",
        xlab = "State", ylab = "Frequency", cex.names = 0.7)
```




```
pie(state_players, col = rainbow(length(state_players)),
    main = "Pie Chart of Home States with Greater than 100 Baseball Players")
```



Answer:

The plot is misleading because it only includes 16 states of a total of 127 states (and cities in other countries), making it seem as if these states are the only birth states of baseball players, and the proportions of players from these states are significantly higher. To address this, we could create an "other" category including the total count from all of the excluded states, which would be one portion of the pie chart or one bar on the bar plot. This would make the percentages labeled in the pie chart accurate.

Problem 4: For loops (10 points)

As discussed in class, for loops allow you to repeat a process many times. Each time the process is repeated, a counter index object (usually named i) is incremented by 1. This is useful because it allows you to:

1. Repeat a process many times to generate results each time
2. Store each result in a vector using i to index into the vector.

The code below create uses a for loop to store the values of 1 squared up to 50 squared in a vector object named my_vec. Modify the code so that what is stored in the vector are the even integers from 2 to 100 (i.e, 2, 4, 6, . . . , 100).

```
my_vec <- NULL
for (i in 1:50){
  my_vec[i] <- i*2
```

```
}

my_vec
```

```
##  [1]   2   4   6   8  10  12  14  16  18  20  22  24  26  28  30  32  34
## [18]  36  38  40  42  44  46  48  50  52  54  56  58  60  62  64  66  68
## [35]  70  72  74  76  78  80  82  84  86  88  90  92  94  96  98 100
```

Problem 5: Short reading (5 points)

As discussed in class, OkCupid is a dating website. One of the founders of the website, Christian Rudder, created a series of blog posts around 2010 where he analyzed data from the site to extract insights about dating. In order gain insight into what is possible from simple descriptive statistics and plots, please read the blog entry from July 7th 2010 title 'The Big Lies People Tell In Online Dating' and write one paragraph comment on something interesting you found in the article. Alternatively, you can read and comment on the article title 'How a Math Genius Hacked OkCupid to Find True Love' and comment on that article instead.

Describe something interesting you found in one of these articles:

After reading the first article published by OkCupid, I was unsurprised by many of their findings. The points published were all occurences I thought would be very common (adding height, exaggerating income, and using old pictures). One aspect I did find interesting was the method which the website used to draw their conclusions. For example, the blog post concluded that men on OkCupid are two inches shorter in reality than they list online by comparing the height distribution of U.S. men to the distribution of OkCupid users. While using the U.S. population to draw comparisons may be the most easily accessible method, I do not think these two data sets can be used to draw the conclusion they did (that the average male exaggerates by 2 inches). There is no indication that the average user exaggerates by a certain amount, and no comparison with their height in reality. I understand this data may be hard to collect, but without it we can only conclude that the OkCupid user is 2 inches taller than the average male.

Reflection (5 points)

Please reflect on how the homework went. In particular, please answer the following questions:

1. What concepts do you feel you are clearly understanding and which concepts are you confused about?
2. How many hours did you spend working on the homework?
3. How much did you enjoy doing the homework ("Super fun", "kind of fun", "not really", or "terrible")?
4. How much do you feel you learned doing this homework ("learned a lot", "learned some", "learned nothing", or "even more confused")?
5. Please note also if you went to TA office hours for help with this worksheet, and if the help you got was useful (in general, we strongly encourage you to attend TA office hours if you are having any difficulties with the homework).
6. Anything else you would like us to know?

Reflection Answers:

1. For me, the coding on this assignment is very clear and easy to understand. All of the concepts in the homework assignment were already covered in class. However, I am having trouble remembering statistics concepts and using statistical terms to describe the data, even though the knowledge required for this assignment was still very basic.
2. About 2.5 hours
3. Kind of fun!
4. Learned some - it was good to review and apply the concepts and code we went over in class, otherwise I wouldn't have been able to retain the information.
5. I got help from Duda, which helped me address a few details I missed (such as labeling or sorting tables).
6. Thank you Prof. Meyers - I'm enjoying the class so far and feel that it is moving at a reasonable pace. The lectures are clear and the lecture slides are even more helpful.

Homework 2

Megan Zhang

The purpose of this homework is to explore sampling distributions, to practice using the bootstrap to construct confidence intervals, and to gain more experience programming in R. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday September 15th. **Note: you might find this homework is more challenging than the previous homework so please get started early.**

If you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Some tips for completing this homework are:

1) Make sure you conceptually understand the problems first before trying to write code for the solutions, i.e, if the problem is asking you to create a plot, draw a picture of the plot and think about the steps needed to get to the answer before writing down any code.
2) Several of the problems ask you to repeat previous problems with different parameter values. The best way to solve this is to do a careful job on the initial problem (e.g., label all the axes well, etc) and then copy your code over and adjust your parameters/answers (and sometimes it's possible to use a for loop over different parameter values to save on writing code).
3) Looking at the notes from class should be helpful

Some useful LaTeX symbols for the problem set are: μ, σ, $\bar{x}$, $\frac{a}{b}$

Problem 1: Exploring sampling distributions with simulations

As discussed in class:

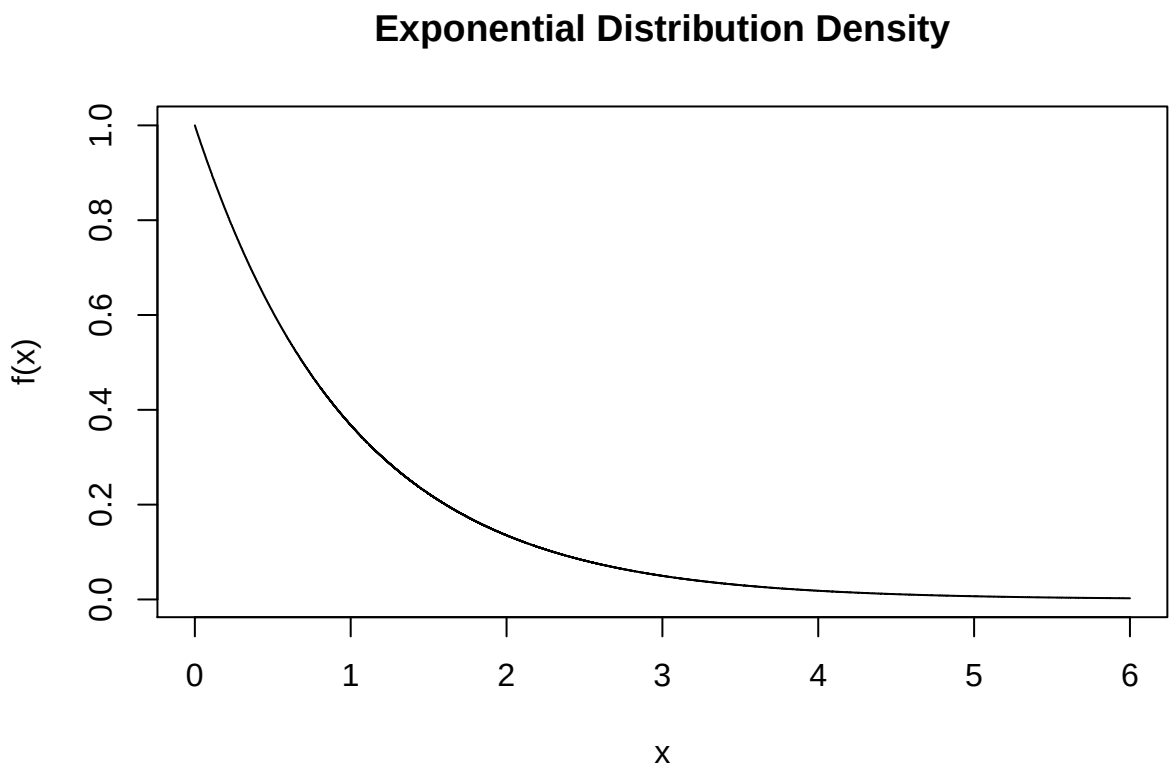
- A **statistic** is a number computed from a sample of data
- A **sampling distribution** is a probability distribution of a *statistic*; i.e., if we repeatedly drew samples of size n from some underlying distribution, and computed the same statistic on each sample, the distribution of these *statistics* is the *sampling distribution*.

The shape of the underlying distribution of data, and the shape of the sampling distribution for a statistic calculated from samples of data, are often quiet different. Below we explore this through simulations.

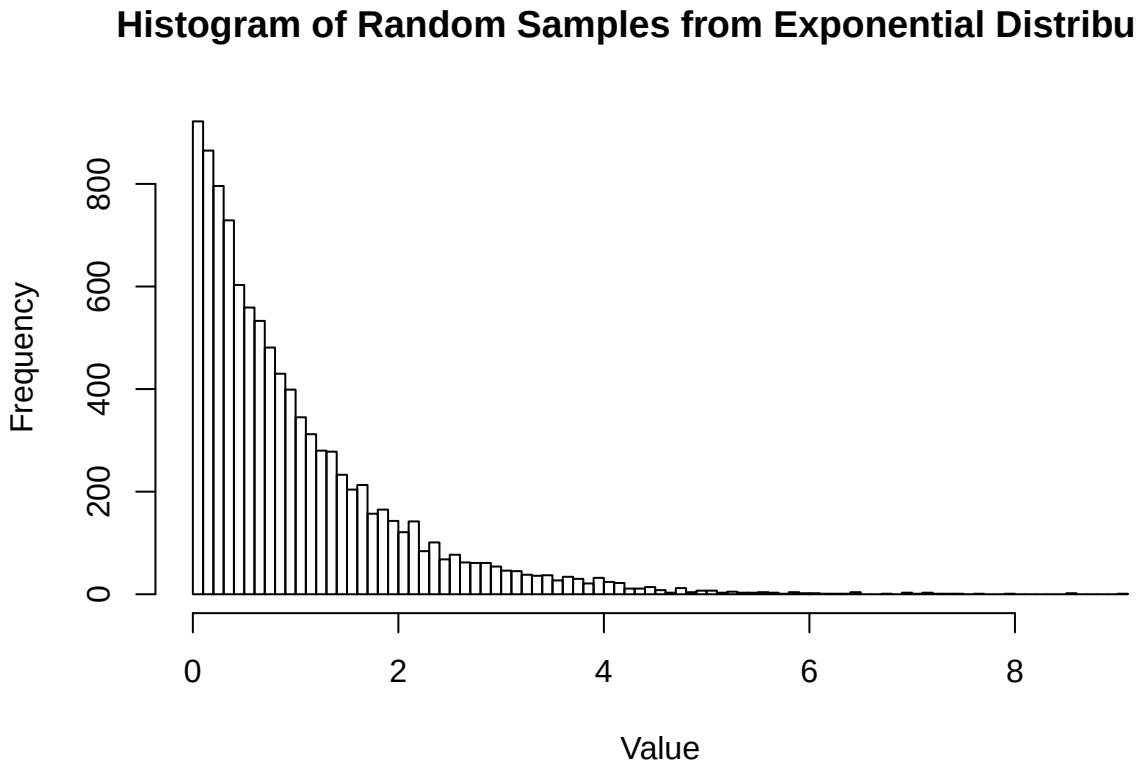
Problem 1.1: (15 points)

Let us examine data that comes from an exponential distribution with rate parameter $\lambda = 1$. Start by plotting the density for this exponential distribution using the dexp() function. Next randomly sample n = 10,000 points from the exponential distribution using the rexp(n) function and plot a histogram of these data (be sure to adjust the nclass argument to bin the histogram more finely). Also, calculate the mean, median and standard deviation of this randomly drawn data, and report the values of these statistics below using the LaTeX for the proper notation (use m for the notation for the median statistic). Finally, discuss whether the values of these statistics are what you would expect based on the values of the parameters of the exponential distribution (looking at the wikipedia entry to learn more about the parameters in the exponential distribution could be useful).

```
# plot the standard exponential density function
x <- seq(0, 6, by = .0001)  # x-values for plotting the exponential density function
y <- dexp(x)
plot(x,y, type = 'l', ylab = "f(x)", main = "Exponential Distribution Density")
```




```
# plot a sample of n = 10,000 points from this distribution
n <- 10000
hist(rexp(n), nclass = 100, xlab = "Value", ylab = "Frequency",
     main = "Histogram of Random Samples from Exponential Distribution")
```




```
# calculate some statistics from this sample
the_mean <- round(mean(rexp(n)), digits = 3)
the_median <- round(median(rexp(n)), digits = 3)
the_sd <- round(sd(rexp(n)), digits = 3)
```

Answers
The following are the statistics of central tendency:
Mean $\bar{x} = 1.003$
Median $m = 0.693$
Standard Deviation $s = 0.99$

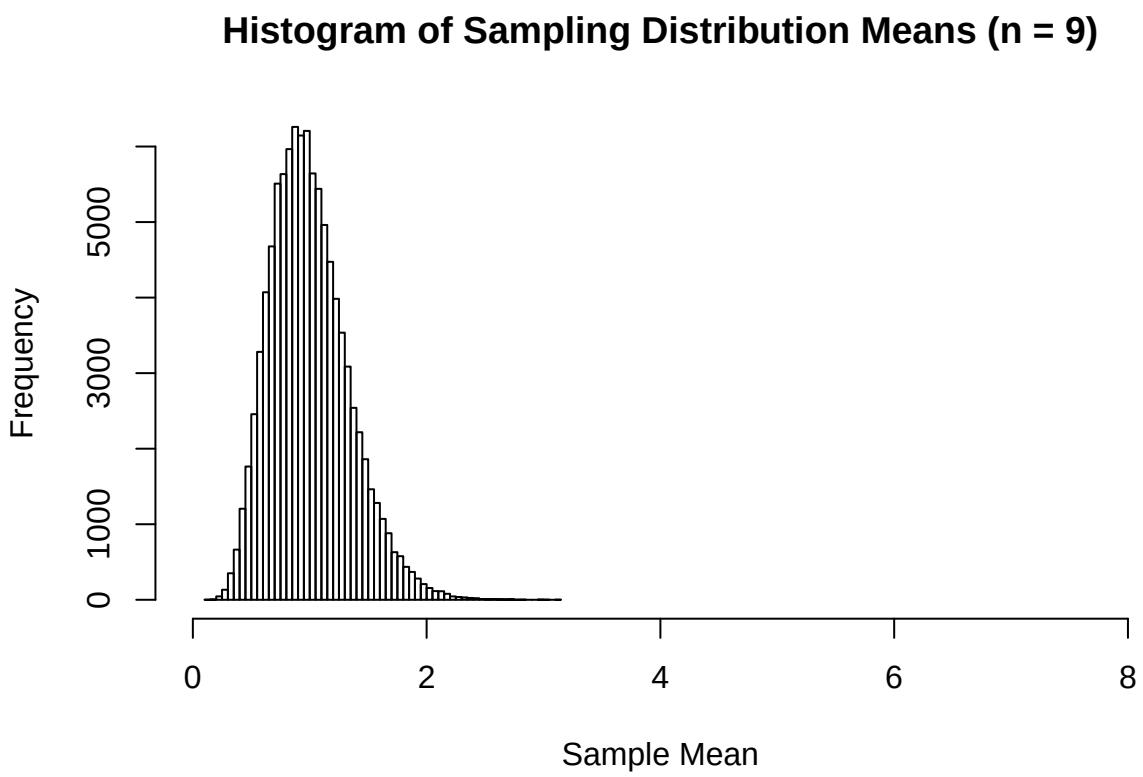
The mean is about 1.00, median 0.693, standard deviation is also about 1.00. This matches the values based on the given parameters: Mean $=1\backslash\lambda = 1\backslash 1 = 1$, Median $= \ln(2)\backslash\lambda = \approx 0.693$, and standard deviation $= \sqrt{variance} = \sqrt{1\backslash\lambda^2} = 1$.

Problem 1.2: (15 points)

Now let's examine the *sampling distribution* for the mean statistic $\bar{x}$ when our underlying distribution is the exponential distribution. Use a for loop to create a sampling distribution that has 100,000 mean statistics, $\bar{x}$, using $n = 9$ points in each sample. Plot the distribution by creating a histogram of these sample statistic values, and set limits on the x-axis to be similar to those of the data distribution in problem 1.1 using the

xlim = c(lower_lim, upper_lim) argument. Finally, describe the shape of this distribution and report the standard error of this distribution.

```
sampling_dist <- NULL
for (i in 1:100000){
  sampling_dist[i] <- mean(sample(rexp(9)))
}

hist(sampling_dist, xlim = c(0,8), main = "Histogram of Sampling Distribution Means (n = 9)",
     xlab = "Sample Mean", ylab = "Frequency", nclass = 50)
```




```
the_SE <- round(sd(sampling_dist), 3)
the_SE
```

```
## [1] 0.334
```

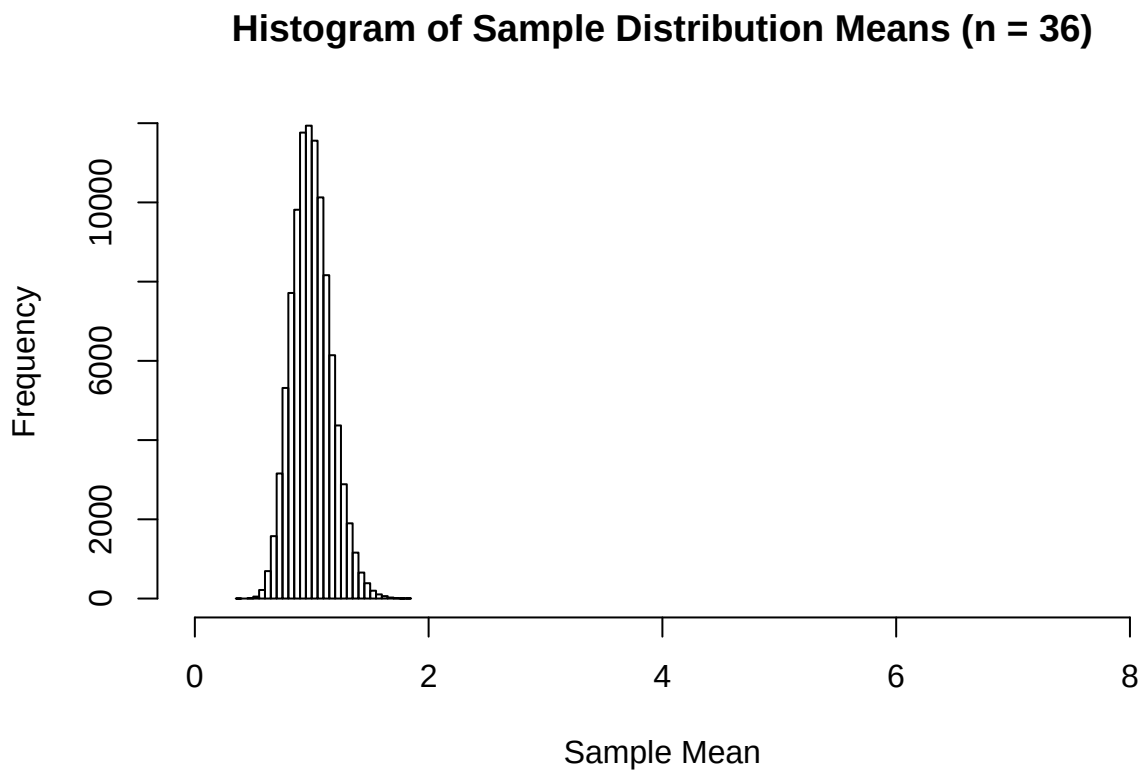
Answers:

The distribution of the sample means is skewed right (the mean is larger than the median). The standard error is SE = 0.334.

Problem 1.3 (15 points)

Now repeat problem 1.2 using sample sizes of n = 36 and n = 144. Report the standard errors for $n = 9$, 36, and 144, and describe how the relationship between values for the standard error SE change with the different values of n. Also describe why it makes sense the SE would get smaller as n increases. Finally describe theoretical results (i.e., a formula) from intro stats that can account for the relationship between between the SE and n (hint: when you have an exponential distribution with rate parameter $\lambda = 1$, the standard deviation of this distribution is $\sigma = 1$).

```
sampling_dist <- NULL
for (i in 1:100000){
  sampling_dist[i] <- mean(sample(rexp(36)))
}

hist(sampling_dist, xlim = c(0,8), main = "Histogram of Sample Distribution Means (n = 36)",
     xlab = "Sample Mean", ylab = "Frequency", nclass = 30)
```

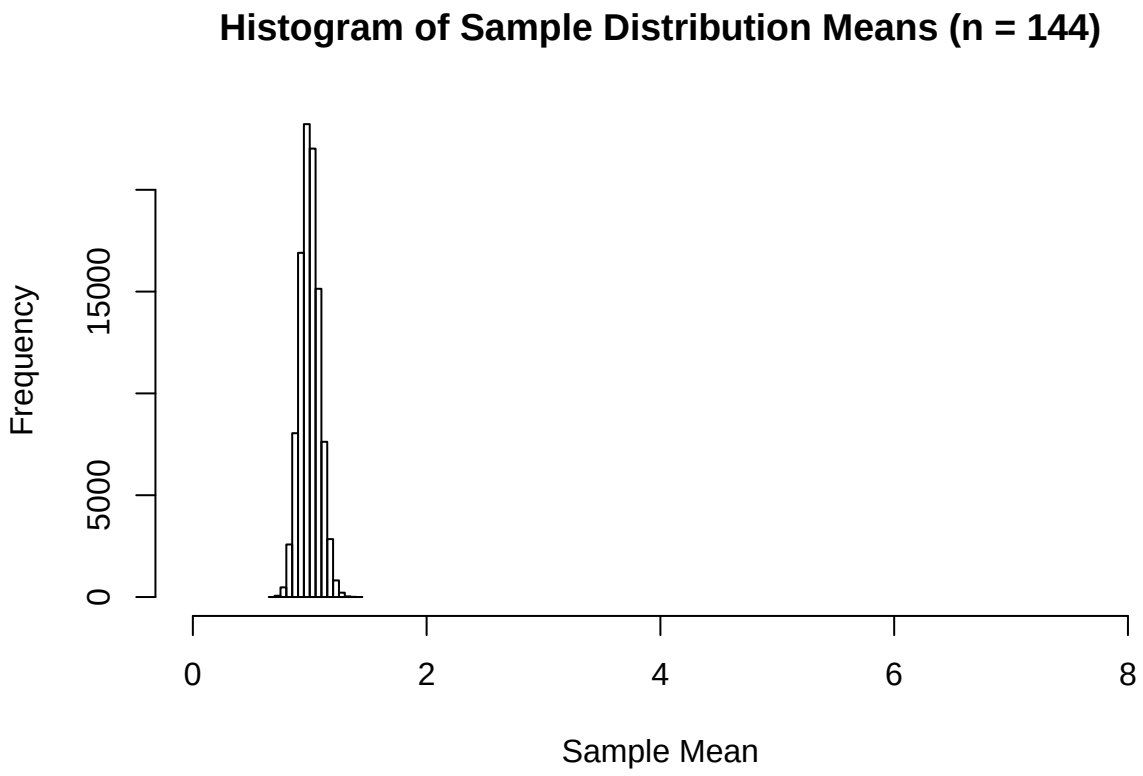



```
the_SE_1 <- round(sd(sampling_dist), 3)
the_SE_1
```

```
## [1] 0.167
```

```
sampling_dist <- NULL
for (i in 1:100000){
  sampling_dist[i] <- mean(sample(rexp(144)))
```

```
}

hist(sampling_dist, xlim = c(0,8), main = "Histogram of Sample Distribution Means (n = 144)",
     xlab = "Sample Mean", ylab = "Frequency", nclass = 20)
```




```
the_SE_2 <- round(sd(sampling_dist), 3)
the_SE_2
```

```
## [1] 0.083
```

Answer:

As n increases, the standard error decreases. This makes sense because according to the Law of Large numbers, as the sample size n increases the sample mean will converge to its expected value. This means the variance among sample means will decrease as they all become close to the true mean. We see this in the above histograms: as sample size increases, the means of the sample distribution become both more normally distributed and vary less. We also see this with our formula for standard error of the mean, which is $\sigma_M = \frac{\sigma}{\sqrt{N}}$, where n is the population size. As we increase the population of the sample, the standard error of the mean will decrease.

SE (n = 9) = 0.334.

SE (n = 36) = 0.167.

SE (n = 144) = 0.083.

Problem 2: Exploring bias correction in the formula for the variance statistic

In intro stats class you learned that the formula for the sample variance statistic is

$$s^2 = \frac{\Sigma_i^n (x_i - \bar{x})^2}{n-1}$$

One question that is often asked by students is why is the denominator in this formula n - 1 rather than just n. To examine this, let's create a sampling distribution of the variance statistic using a denominator of n - 1 and compare it to using a denominator of n.

Part 2.1 (10 points)

The function var() calculates the variance statistic from a data sample. Also, the function written below called var_n calculates the variance using a denominator of n rather than n-1. Create a sampling distribution using var() and var_n() when data comes from the standard normal distribution (using rnorm) for a sample size of n = 10. Plot histograms of these sampling distributions, and calculate the mean of these sampling distributions. Also use the abline(v = ...) function to plot a vertical line at the value of the parameter $\sigma^2 = 1$ (in red), and the value for the mean (expected value) of the sampling distribution (in blue). Then report below:

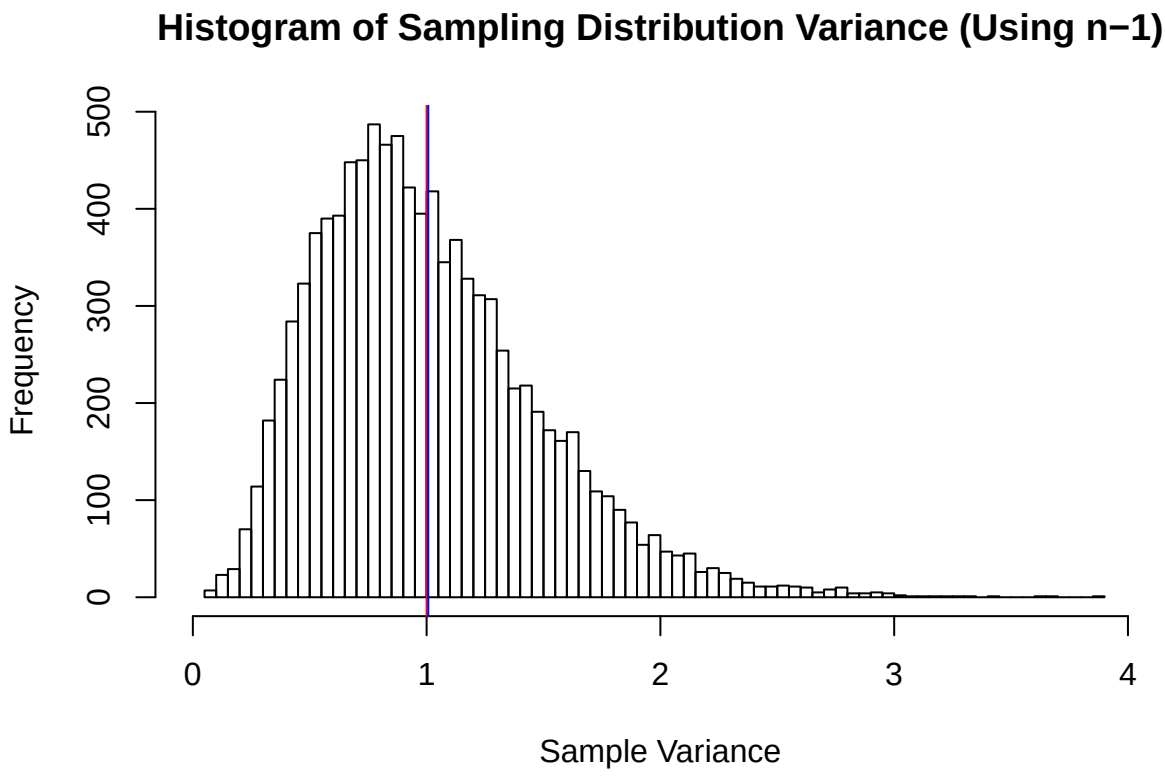
1) The shapes of these distributions
2) Whether the means of these sampling distribution equal the underlying variance parameter of $\sigma^2 = 1$.

Note: a statistic (i.e., estimator) is called *biased* if it's mean (expected value) does not equal the population parameter it is trying to estimate. Thus if the mean value of our sampling distribution does not equal the population parameter (in this case $\sigma^2 = 1$) then our statistic (estimator) is biased.

```
var_n <- function(data_sample){
  var(data_sample) * (length(data_sample) - 1)/length(data_sample)
}


# continue from here...
sampling_dist_1 <- NULL
for (i in 1:10000){
  sampling_dist_1[i] <- var(rnorm(10))
}

hist(sampling_dist_1, main = "Histogram of Sampling Distribution Variance (Using n-1)",
     xlab = "Sample Variance", ylab = "Frequency", nclass = 75, xlim = c(0,4))
mean_var <- mean(sampling_dist_1)
mean_var
```

```
## [1] 1.007032
```

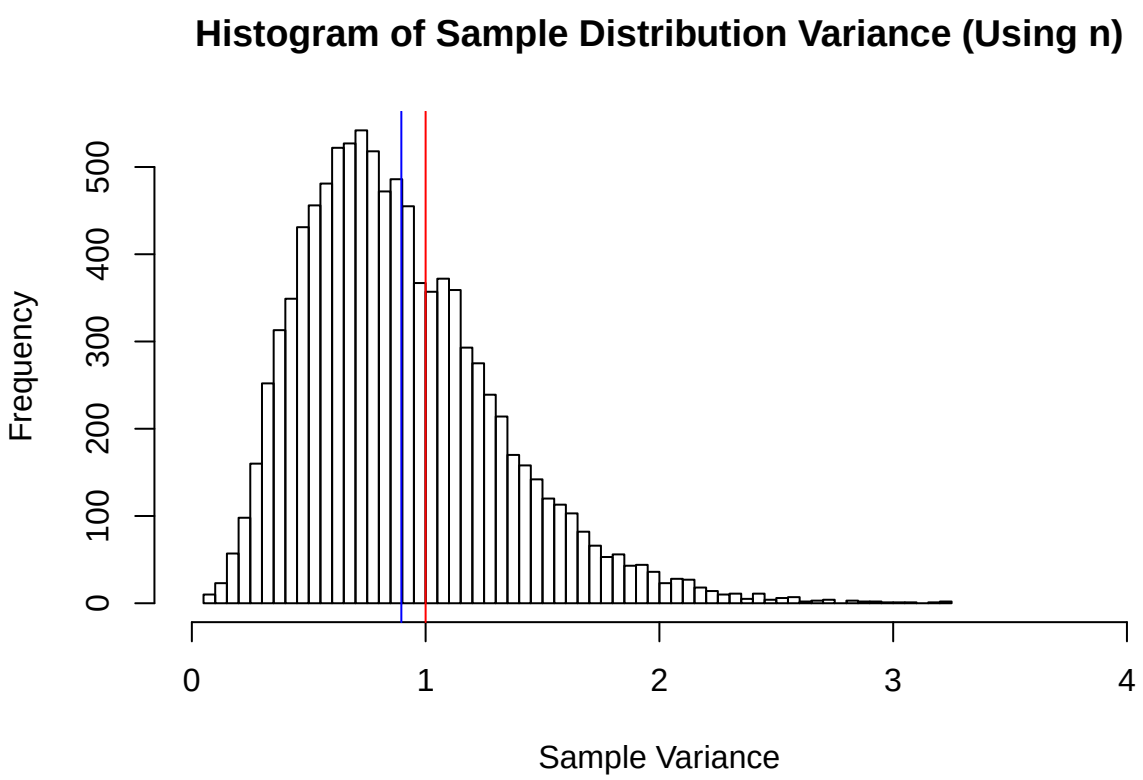
```
abline(v = 1, col = "red")
abline(v = mean_var, col = "blue")
```




```
# continue from here...
sampling_dist_2 <- NULL
for (i in 1:10000){
  sampling_dist_2[i] <- var_n(rnorm(10))
}

hist(sampling_dist_2, main = "Histogram of Sample Distribution Variance (Using n)",
     xlab = "Sample Variance", ylab = "Frequency", nclass = 75, xlim = c(0,4))
mean_varn <- mean(sampling_dist_2)
mean_varn
```

```
## [1] 0.8961534
```

```
abline(v = 1, col = "red")
abline(v = mean_varn, col = "blue")
```



Answers:

Both are skewed right, but only mean of the distrubution using the function var() (n-1 is the denominator) is equivalent to the variance of 1. Here the blue and red lines are overlapping when we use n as the denominator to calculate variance. However, the variance using n as the demoninator is smaller than 1, so this sampling distribution is biased.

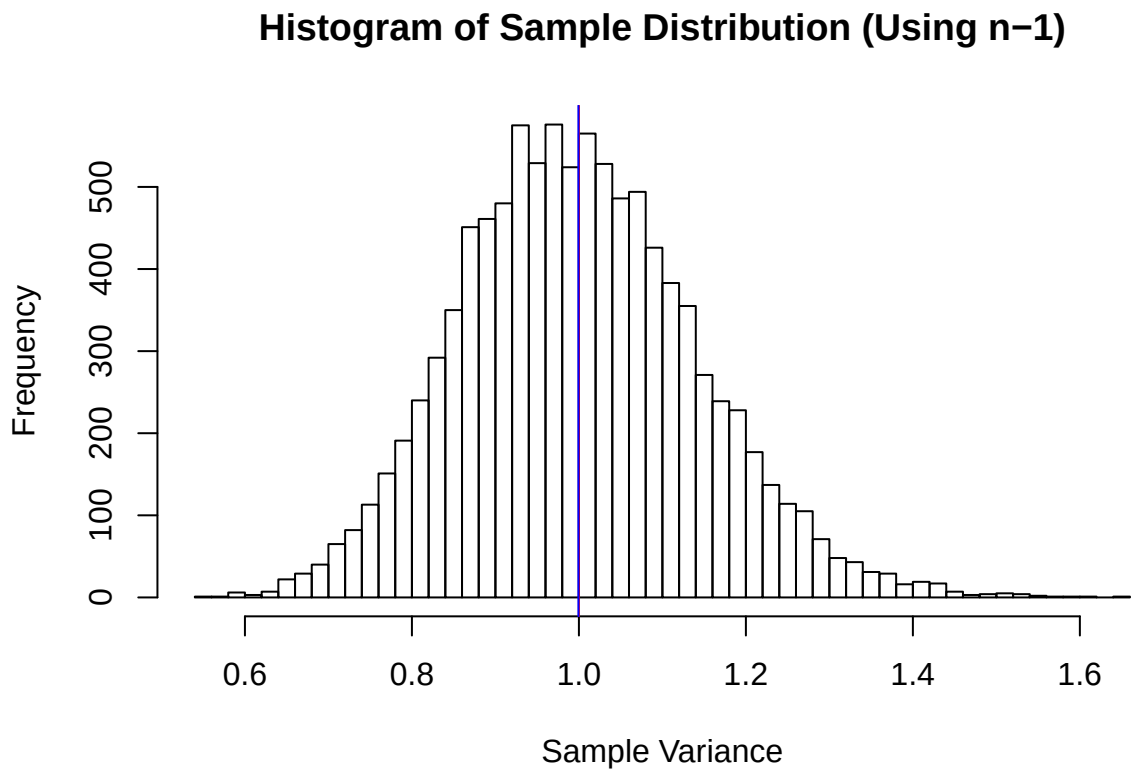
Part 2.2 (10 points)

Repeat part 2.1 but using a sample size of n = 100. Do the answers to the questions posed in part 2.1 change?

```
sampling_dist_1 <- NULL
for (i in 1:10000){
  sampling_dist_1[i] <- var(rnorm(100))
}

hist(sampling_dist_1, main = "Histogram of Sample Distribution (Using n-1)",
     xlab = "Sample Variance", ylab = "Frequency", nclass = 75)
mean_var <- mean(sampling_dist_1)
mean_var
```

```
## [1] 0.9994331
```

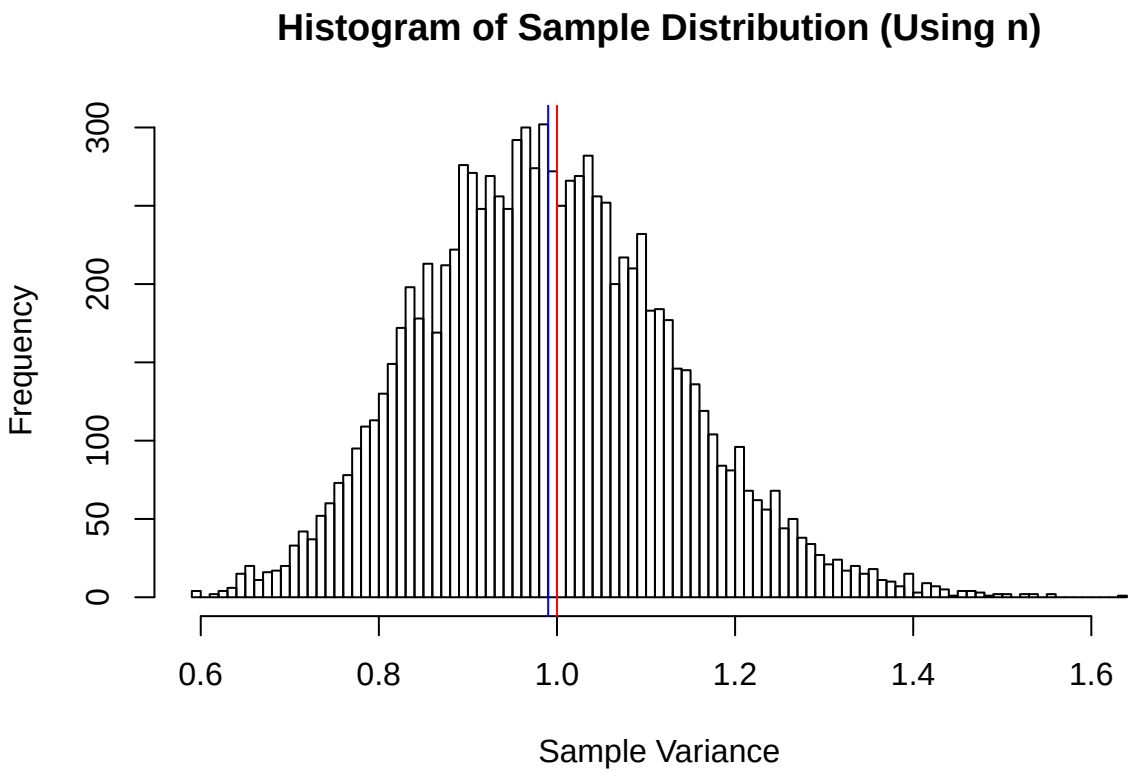
```r
abline(v = 1, col = "red")
abline(v = mean_var, col = "blue")
```




```r
# continue from here...
sampling_dist_2 <- NULL
for (i in 1:10000){
  sampling_dist_2[i] <- var_n(rnorm(100))
}

hist(sampling_dist_2, main = "Histogram of Sample Distribution (Using n)",
     xlab = "Sample Variance", ylab = "Frequency", nclass = 75)
mean_varn <- mean(sampling_dist_2)
mean_varn
```

```
## [1] 0.9901438
```

```r
abline(v = 1, col = "red")
abline(v = mean_varn, col = "blue")
```



Answers:

Adding to the sample causes the distribution to become more normally distributed. This is explained by the Central Limit Theorem. The sample variance calculated using n as the denominator is still less than the mean, but there is less difference due to the larger sample size.

Reflection (5 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 2

Homework 3

The purpose of this homework is to practice using the bootstrap to construct confidence intervals, and to learn how to use randomization methods to run hypothesis tests. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday September 22nd.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

If you want to learn more about the bootstrap (and why it is useful for teaching the concept of confidence intervals), please read this paper by Tim Hesterberg

Problem 1: Calculating confidence intervals using the bootstrap

As discussed in class, we can use the bootstrap to estimate standard errors which can then be used to calculate confidence intervals. Let's use the bootstrap to calculate a confidence interval for the mean height of OkCupid users.

Part 1.1 (10 points)

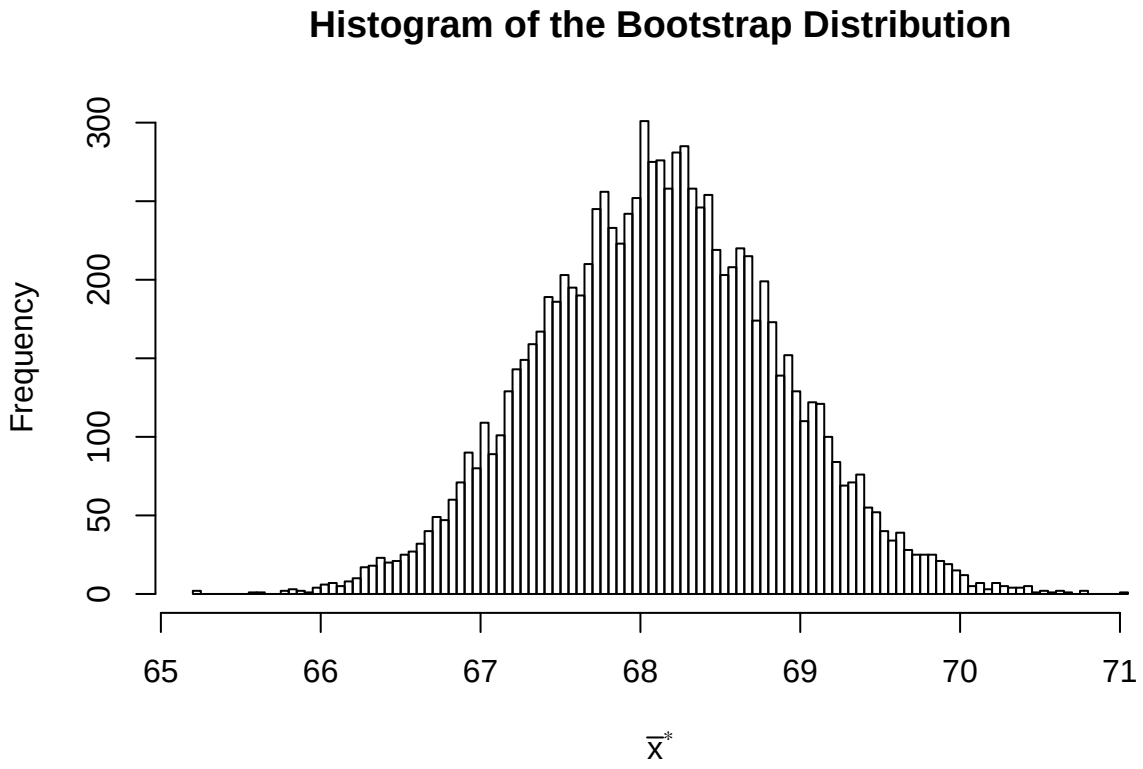
To explore how confidence intervals work when a sample size is relatively small (and also to make this problem more computationally efficient), use the heights from only the first 20 OkCupid users, and then do the following steps:

1) Estimate the standard error of the mean height using the bootstrap
2) Plot a histogram of the bootstrap distribution
3) Calculate an approximate 95% confidence interval for the heights of OkCupid users using the formula CI $[\bar{x} - 2 \cdot SE^*, \bar{x} + 2 \cdot SE^*]$

Report your confidence interval below and describe what the confidnece interval tells you.

```
# load the OkCupid data and extract the heights for the first 20 profiles
library(okcupiddata)
the_heights <- profiles$height[1:20]


# construct the bootstrap distribution
bootstrap_dist <- NULL
for (i in 1:10000){
  boot_sample <- sample(the_heights, replace = TRUE)
  bootstrap_dist[i] <- mean(boot_sample)
}
```

```r
# plot a histogram of the bootstrap distribution
hist(bootstrap_dist,
     nclass = 100,
     xlab = TeX("$\\bar{x}$^*"),
     main = "Histogram of the Bootstrap Distribution")
```




```r
# calculate the standard error
(SE <- sd(bootstrap_dist))
```

```
## [1] 0.7496106
```

```r
# calculate 95% confidence intervals
CI_lower <- mean(the_heights) - 2 * SE
CI_upper <- mean(the_heights) + 2 * SE

c(CI_lower, CI_upper)
```

```
## [1] 66.65078 69.64922
```

Answer:

The 95% confidence interval is [66.64, 69.66]. This means that there is a 95% chance that this interval contains the true population mean. The standard error SE = 0.749.

Part 1.2 (10 points)

Run your code above again but 1) use the first 100 OkCupid users, and 2) then use the first 1000 OkCupid users. Report what the confidence interval are when using these different number of users (i.e., when using different sample sizes n). Do they seem much smaller? Note: you do not need to show code here, just modify the code above rerun it and then report the results.

Answer:
The 95% confidence interval using the first 100 users is [68.17, 69.63] (Standard error = 0.363). Using the first 1000 users the interval is [68.16, 68.66] with a standard error of 0.125. The interval decreases in size as the sample size increases.

Part 1.3 (10 points)

Now write code to create confidence intervals separately for the heights of male and female OkCupid users by using the subset() function to get separate vectors of heights for males and females and use the first 100 male and 100 female okcupid users. Does it appear plausible that the actual mean height for males μ_{male} is the same as the actual mean height for females μ_{female} ?

```
# get the hights for the first 100 male and 100 female OkCupid users
the_heights_male <- subset(profiles, sex == "m")

# continue from here...
hundred_male <- the_heights_male$height[1:100]

the_heights_female <- subset(profiles, sex == "f")
hundred_female <- the_heights_female$height[1:100]



# create bootstrap distributions for the male and female heights
bootstrap_dist_m <- NULL
for (i in 1:10000){
  boot_sample_m <- sample(hundred_male, replace = TRUE)
  bootstrap_dist_m[i] <- mean(boot_sample_m)
}

bootstrap_dist_f <- NULL
for (i in 1:10000){
  boot_sample_f <- sample(hundred_female, replace = TRUE)
  bootstrap_dist_f[i] <- mean(boot_sample_f)
}


# calculate standard errors and confidence intervals for the male and female heights
(SE_m <- sd(bootstrap_dist_m))
```

```
## [1] 0.2715646
```

```r
(SE_f <- sd(bootstrap_dist_f))
```

```
## [1] 0.2444839
```

```r
# calculate 95% confidence intervals
(CI_male <- c((mean(hundred_male) - 2 * SE_m),
              (mean(hundred_male) + 2 * SE_m)))
```

```
## [1] 69.99687 71.08313
```

```r
(CI_female <- c((mean(hundred_female) - 2 * SE_f),
              (mean(hundred_female) + 2 * SE_f)))
```

```
## [1] 64.56103 65.53897
```

Answers: The 95% Confidence Interval for male heights is [69.99, 71.09] with a standard error of 0.276. The 95% Confidence Interval for female heights is [64.56, 65.54] with a standard error of 0.276. Since these intervals do not overlap, it does not seem plausible that $\mu_{male} = \mu_{female}$.

Problem 2: Comparing bootstrap CIs to formula based CIs

As you (almost certainly) learned in Introduction to Statsistics, there is a mathematical forumla that can give you the standard error for the mean statistic $\bar{x}$ (note: standard error for the mean statistic is called "the standard error of the mean" and is often abbreviated SEM or denoted as $\sigma_{\bar{x}}$ or $s_{\bar{x}}$).

The formula for the SEM is:

(1) $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$

and an estimate for this is given by:

(2) $s_{\bar{x}} = \frac{s}{\sqrt{n}}$

Where:

- σ is the population standard deviation,
- s is the standard deviation statistic computed from a sample of size n
- n is the sample size.

Note, equation 1 above is a theoretical construct since we will never know σ (only Plato knows this) while equation 2 above is possible to calculate from a sample of data.

Part 2.1 (10 points)

Using the formula in equation 2 above, repeat the analyses in problem 1.1 by calculating the standard error of the mean, and a 95% confidence interval for the mean height of OkCupid users, but use formula 2 to caluclate the standard error rather than the bootstrap. Again, use only the first 20 OkCupid users in the data set. Is the confidence interval you created using formula 2 close to the confidence interval you created in problem 1.1?

```r
# calculate the SEM and CI using the formula in equation 2
the_heights <- profiles$height[1:20]
(SE_eq <- (sd(the_heights))/sqrt(20))
```

```
## [1] 0.7687139
```

```r
(CI_eq <- c((mean(the_heights) - 2*SE_eq),
            (mean(the_heights) + 2*SE_eq)))
```

```
## [1] 66.61257 69.68743
```

The 95% confidence interval is [66.61, 69.69] with a standard error of 0.769. The mean remains the same, but the standard error has increased by 0.02, resulting in a wider interval.

Answers:

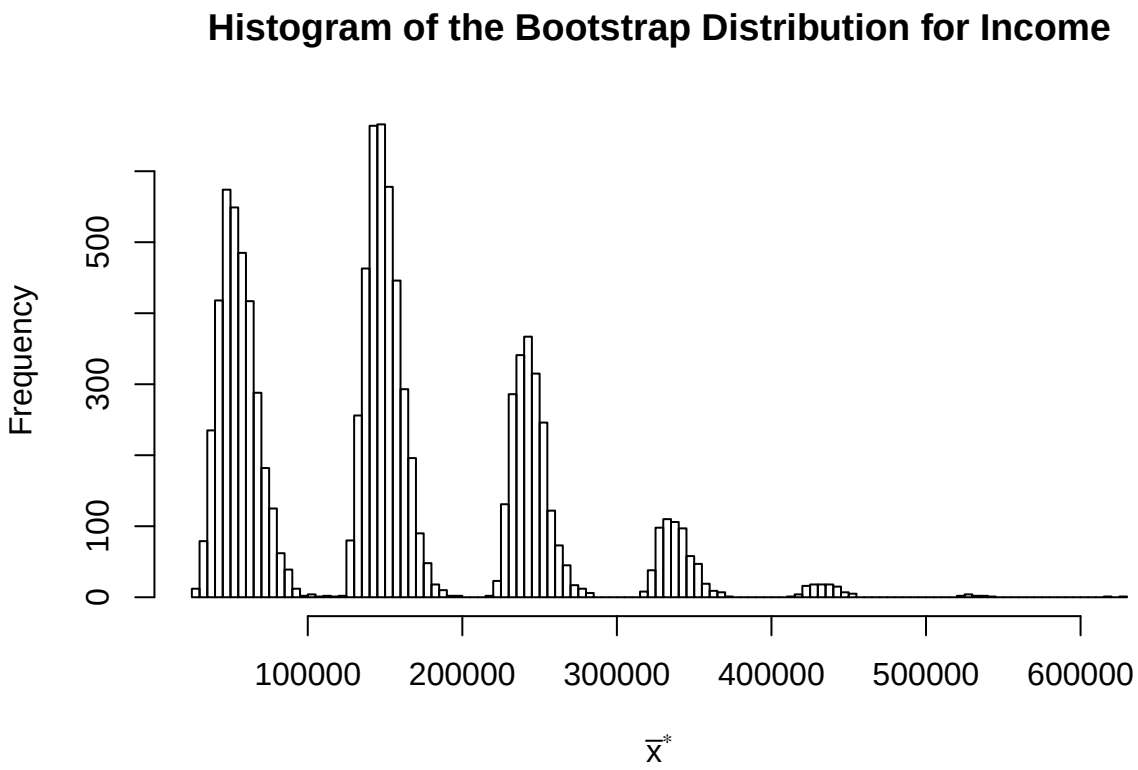
Part 2.2 (15 points)

The line of code below extracts the incomes from the first 10 OkCupid users. Compare the confidence intervals for the mean income using:

1) the formula for the standard error
2) the bootstrap estimate of the standard error
3) the bootstrap percentile method

Report which confidence interval(s) seems to give the most reasonable results, and give an explanation for why they differ. Also, assuming that the whole population is just the OkCupid users in the profiles data frame, calculate the value of the parameter that these confidence intervals are trying to capture (using the appropriate symbol) and report whether all these methods capture this parameter value.

```r
# extract OkCupid user's incomes from the users that listed their income
the_income <- na.omit(profiles$income)[1:10]


# create a bootstrap distribution
bootstrap_dist_income <- NULL
for (i in 1:10000){
  boot_sample_income <- sample(the_income, replace = TRUE)
  bootstrap_dist_income[i] <- mean(boot_sample_income)
}

# plot the boostrap distribution

hist(bootstrap_dist_income,
     nclass = 100,
     xlab = TeX("$\\bar{x}$^*"),
     main = "Histogram of the Bootstrap Distribution for Income")
```




```
# calculate the bootstrap estimate of the standard error SE*
(SE_inc <- sd(bootstrap_dist_income))
```

```
## [1] 90315.51
```

```
# calculate a CI using the bootstrap perentiles
(CI_boot_percentile <- quantile(bootstrap_dist_income, c(.025, .975)))
```

```
##   2.5%  97.5%
##  39000 346000
```

```
# calculate the CI based on using equation 2 to estimate the SE
(CI_boot_SE <- c(mean(the_income) - 2 * SE_inc, mean(the_income) + 2 * SE_inc))
```

```
## [1] -30631.02 330631.02
```

```
# calculate the parameter value
mean(the_income)
```

```
## [1] 150000
```

```r
mean(na.omit(profiles$income))
```

```
## [1] 104395
```

```r
#calculate using the fomula
(SE_form <- sd(the_income)/sqrt(10))
```

```
## [1] 95207.38
```

```r
(CI_boot_formula <- c(mean(the_income) - 2 * SE_form, mean(the_income) + 2 * SE_form))
```

```
## [1] -40414.75 340414.75
```

Answers

As we can see from the histogram of the bootstrap distribution, there are multiple peaks at which a particular sample mean value will occur at most often. This means the original sample contains a wide distribution of values, and since the sample size is small, the value of the mean can vary drastically depending on which elements of the sample are chosen. This also means the sample error of the distribution will be very large, which causes the confidence interval to be too large, containing negative values. The problem with the bootstrap distribution and formula methods is that they assume the sample is normally distributed, so the 95% confidence interval will hold. However, our sample is not normally distributed and thus it is more logical to use the bootstrap percentile, which makes calculations based off the values which are actually in the set.

Reflection (5 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 3

Homework 4

The purpose of this homework is to practice using randomization methods to run hypothesis tests. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:30pm on Sunday September 29th.

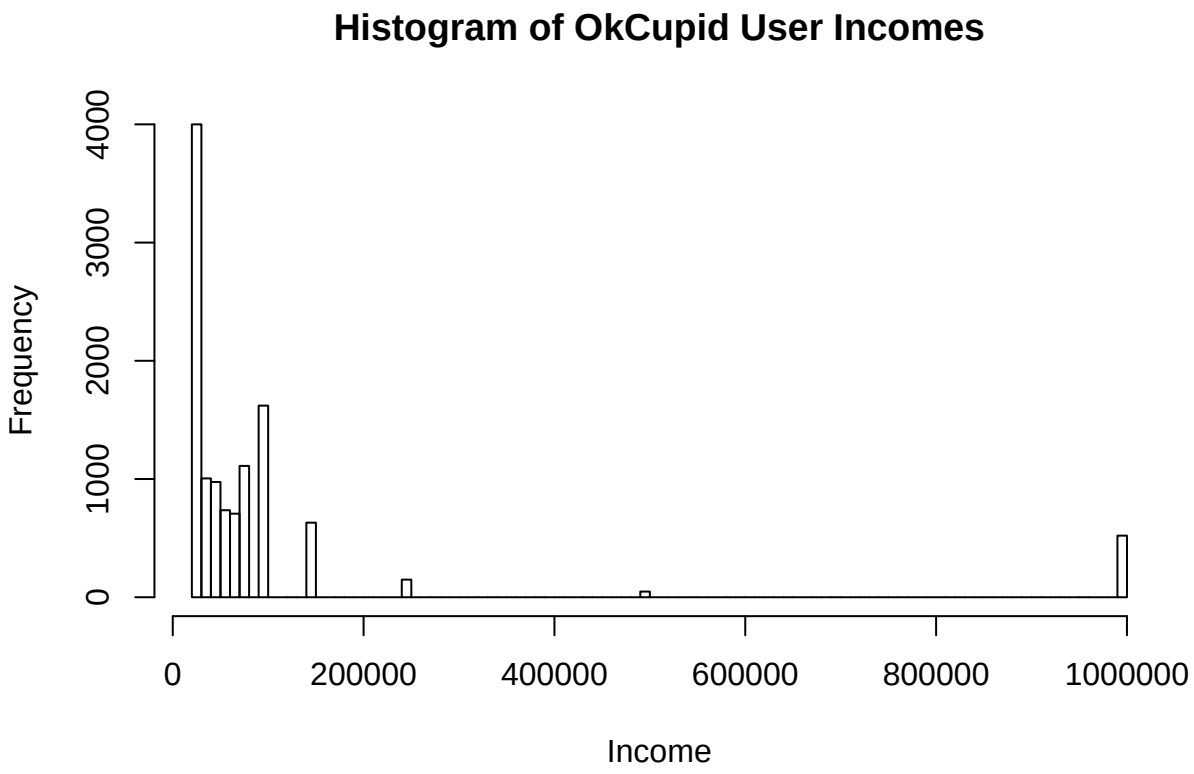
Part 1: Further exploration of OkCupid users' income

Part 1.1 (5 points) In homework 3 (problem 2.2) you calculated the population mean income for OkCupid μ assuming that the population consisted of only OkCupid users in the profiles data frame (and only those users who reported their income). The value for the population parameter for the mean income μ that you got should have been around $\$100,000$. The question then becomes, do we really believe that a typical OkCupid users is making around $\$100k$ a year?

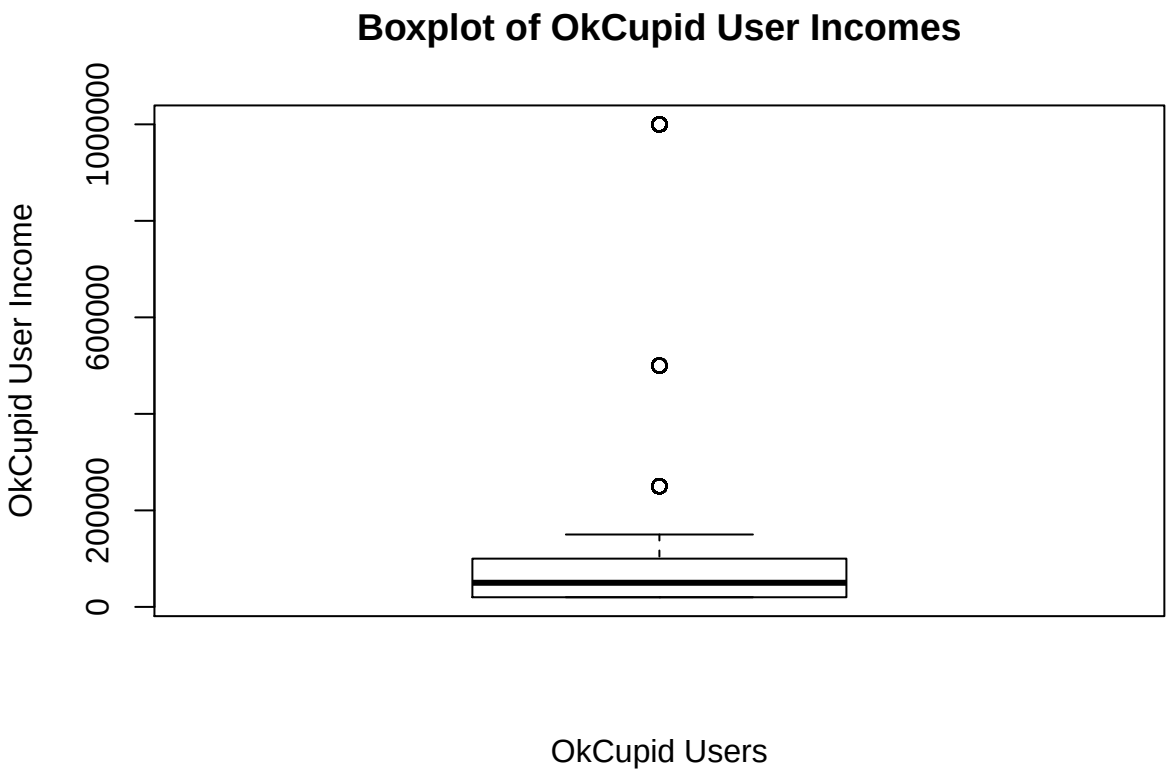
Before diving into inferential statistics (such as creating confidence intervals and running hypothesis test) we really should always explore and visualize the data first. So let's do some exploratory analysis now, which will also be a useful review of some of the material we have already covered in this class.

To start, create a histogram and a boxplot of the income data from the OkCupid users. Also, calculate the proportion of users who claimed their income to be a million dollars or more, and report the value below. Does the portion of people who say they are making a million dollars or more match what is reported in surveys of Americans? (use internet sources to get an estimate of how many Americans make a million dollars or more a year).

```
library(okcupiddata)

all_income <- na.omit(profiles$income)
hist(all_income, main = "Histogram of OkCupid User Incomes",
     xlab = "Income", ylab = "Frequency", nclass = 100)
```




```
boxplot(all_income,
xlab = "OkCupid Users",
ylab = "OkCupid User Income",
main = "Boxplot of OkCupid User Incomes")
```




```
#Proportion of million-dollar earners
(million_prop <- sum(all_income >= 1000000)/length(all_income))
```

```
## [1] 0.0452886
```

Answer

According to this data, 4.5% of OkCupid users are million-dollar earners. According to data from the university of Minnesota, in 2018 the 99th percentile of income in the United States was around $\$300,000$. This means less than 1% of people in the United States make more than 1 million a year. This is far below the proportion calculated from the reports of OkCupid users.

Part 1.2 (5 points) Now let's examine how the mean statistic is affected if we remove the users who reported making a milion dollars or more. To do this, compare the population mean income (i.e., mean of all valid values in the whole profiles data frame) when users who are making a million or more are excluded, to the mean income when all users are included, and create side-by-side boxplots of these two populations with the outliers removed from the plots. Report what these mean values are and whether the millionaires have a big impact on our estimates of the mean income. Also, report whether the boxplots look similar.

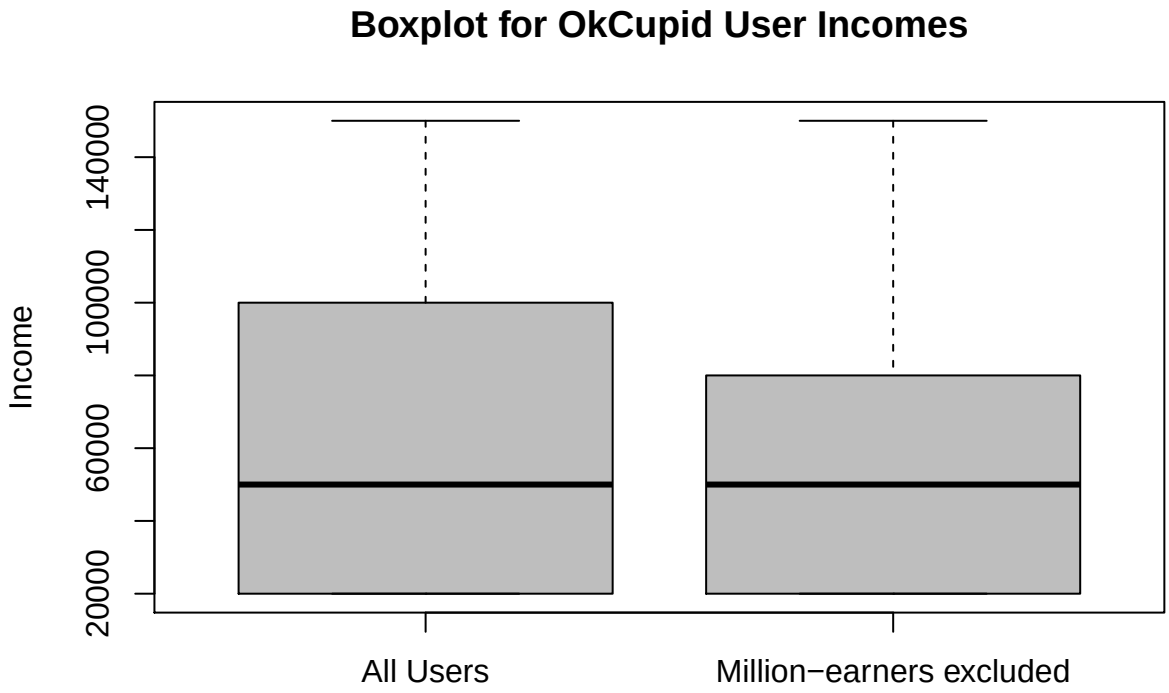
```r
#Mean for all users
all_income <- na.omit(profiles$income)
(mean(all_income))
```

```
## [1] 104395
```

```r
#Mean for users, million earners excluded
million_excl <- all_income[all_income < 1000000]
(mean(million_excl))
```

```
## [1] 61910.22
```

```r
#Comparative Boxplot
boxplot(all_income, million_excl, outline = FALSE, col = "gray",
        main = "Boxplot for OkCupid User Incomes",
        names = c("All Users", "Million-earners excluded"),
        ylab = "Income")
```



Answers

The mean income for all users was $\$104,385$, wheras the mean income for users excluding millionares is $\$61,910.22$. The presence of millionares on the data set does have a large impact on the mean income (the mean was increased about $\$40,000$). However, when we look at the boxplots it seems that they are very similar. The medians are very close, likely because the median is robust to outliers (the removal of the million-dollar values most likely did not change the middle value of the set). There is some difference in that

the third quartile is smaller when millionaires are excluded, which is probably because the third quartile is likely to be affected by the removal of the highest values.

Part 1.3 (10 points) As we saw above, the mean statistic is not very resistant to outliers (i.e. it can be heavily influenced by large values). This can lead to estimates that are not really representative of what we might think of as a "typical American". The median statistic is resistant to large values however, and so might be more meaningful here.

Let's examine the median by doing the following:

1) Calculating the median income on the whole okcupid data set a) including and b) excluding people who report making over one million dollars.
2) Use the bootstrap percentile method to create a confidence interval for the median using the first 50 OkCupid users. Again, do this a) including and b) excluding people who report making over one million dollars. (Note: confidence intervals are always computed on a sample of data (here of size n = 50), and the purpose of this exercise is to see whether our confidence interval based on using the bootstrap percentile method captures the true population median parameter calculated from all OkCupid users).

For the confidence interval, you only need to show your code when including all users (part a's) but fill in the table below reporting the values. Alternatively, create a function for calculating bootstrap confidence intervals and run your code twice with different the million dollar incomes included vs. excluded.

Fill in the table below showing the mean values calculated above from the whole population, as well as the median values and the median confidence intervals based on 50 users, and describe whether the results for the median change much depending on whether the millionares' incomes are excluded. Also describe whether the median seems like a better description of a "typical income" compared to the mean, and if there might be any issues trusting the confidence intervals that were created.

```
#Median of entire dataset
median(all_income)
```

```
## [1] 50000
```

```
#Median of data with exclusion
median(million_excl)
```

```
## [1] 50000
```

```
#Bootstrap percentile function
bts_per_func <- function(income_vec){
bootstrap_dist_income <- c()
for (i in 1:10000){
  bootstrap_dist_income[i] <- median(sample(income_vec[1:50], replace = T))
}
(CI_boot_per <- quantile(bootstrap_dist_income, c(.025, .975)))
}

bts_per_func(all_income)
```

```
##  2.5% 97.5%
## 50000 80000
```

```
bts_per_func(million_excl)
```

```
##  2.5% 97.5%
## 50000 80000
```

Answer

	pop mean	pop median	CI median
all data	104395	50000	(50000, 80000)
exluding millionaire income	61910.22	50000	(50000, 80000)

The population median as well as Confidence Interval do not not change whether or not we remove the millionares in the data set. In this way it seems to be a better descriptor of income compared to the mean, since the median captures the "middle" value of the income without being influenced by the outliers, as the mean statistic would. However, there may be some issues trusting these confidence intervals because we notice that the population median both including and excluding millionaires is only at the end of the interval. Also, taking a sample of only 50 users causes there to be a wider interval since the sample may not accurately represent the entire population.

Problem 2: Hypothesis tests and confidence intervals for a single proportion

Paul the Octopus was an octopus who became famous for predicting winners of soccer matches during the 2010 World Cup. To examine Paul's psychic abilities, two containers of food (mussels) were lowered into the Paul's tank prior to each soccer game. The containers were identical, except for country flags of the opposing teams, one on each container. Whichever container Paul opened first was deemed his predicted winner.

Paul (in a German aquarium) became famous for correctly predicting 11 out of 13 soccer games during the 2010 World Cup. Let's use hypothesis testing to examine whether Paul is actually psychic or if he was merely guessing.

Part 2.1 (5 points): State the null and alternative hypotheses testing whether Paul is psychic using both words and in the appropriate symbols. Also describe what the significance level means and denote it with the commonly used symbol and commonly used value.

Answer:

H_0 : Paul does not have the ability to predict the winners (His guessing ability is $\pi = 0.5$).

H_A : Paul has the ability to predict the winners of soccer matches (His guessing ability is $\pi > 0.5$)

The significance level α, which is commonly $\alpha = 0.05$, is the probability that we reject the null hypothesis when it is actually true. This means there is a 5% risk of concluding Paul is psychic when he is actually not.

Part 2.2 (5 points) : Compute the statistic of interest and save it in an object paul_stat. Do you think it is likely you would get a statistic this extreme if Paul was guessing?

```r
# calculate the observed statistic
(paul_stat <- 11/13)
```

```
## [1] 0.8461538
```

Answer: If Paul was guessing, his success rate should be closer to 0.5, so it seems unlikely that he would have a success rate of $\hat{p} = 0.846$.

Part 2.3 (5 points) : Now use the rbinom() function to generate a null distribution that would occur if Paul was guessing, and save the results in an object called null_distribution.

Remember that the arguments to the rbinom(num_sims, size, prob) are:

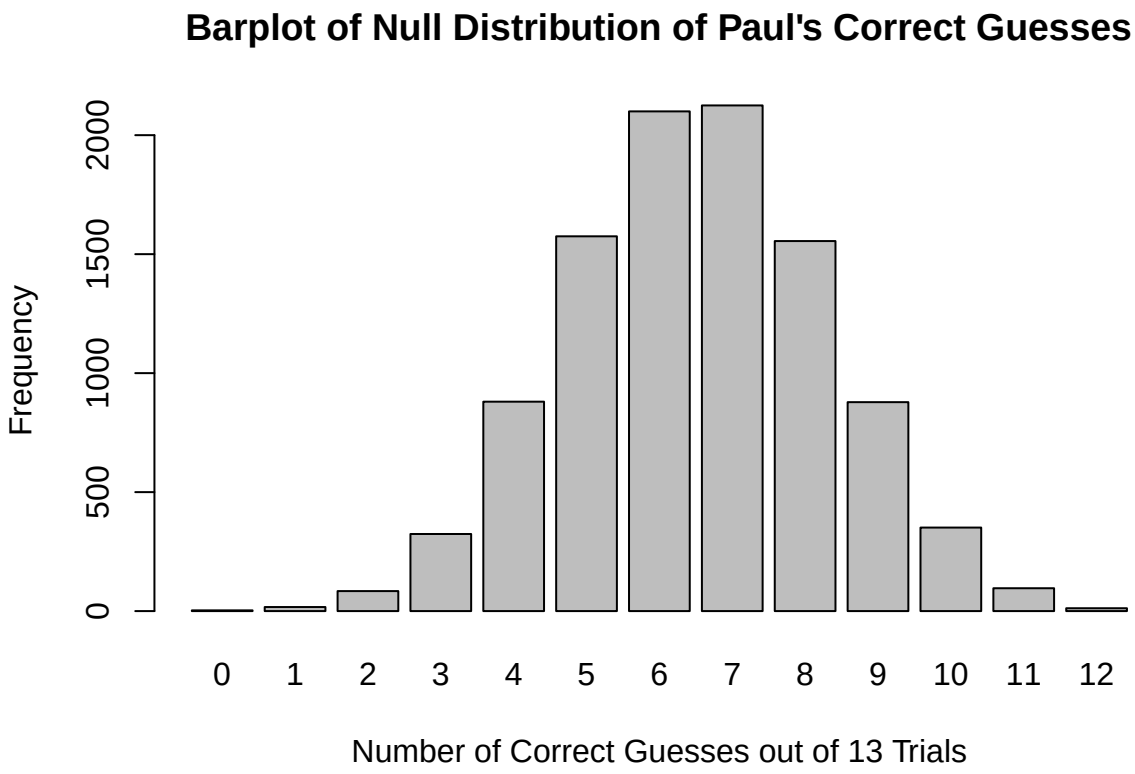
- num_sims: the number of simulations to run
- size: the number of "coin flips" in each simulation
- prob: the probability of getting heads on each coin flip.

Also create a table showing the number of "heads" each simulation produced, and plot this null distribution table as a bar plot.

```r
null_distribution <- rbinom(10000, 13, 0.5)
table(null_distribution)
```

```
## null_distribution
##    0    1    2    3    4    5    6    7    8    9   10   11   12
##    3   17   84  324  880 1575 2100 2125 1555  878  351   96   12
```

```r
barplot(table(null_distribution),
        main = "Barplot of Null Distribution of Paul's Correct Guesses",
        xlab = "Number of Correct Guesses out of 13 Trials", ylab = "Frequency")
```



Part 2.4 (5 points): Now use the variables null_distribution and paul_stat to calculate the number of simulations that had as many or more "heads" than as Paul's correct soccer prediction answers. Convert this to a p-value by dividing by the total number of simulations. Does this p-value provide evidence that Paul is psychic?

```
sum(null_distribution/13 >= paul_stat)
```

```
## [1] 108
```

```
#p-value
(p_value <- sum(null_distribution/13 >= paul_stat)/10000)
```

```
## [1] 0.0108
```

Answer: p-value $= 0.0108$. This is less than $\alpha = 0.05$, therefore the results are statistically significant and we are provided evidence that Paul is psychic.

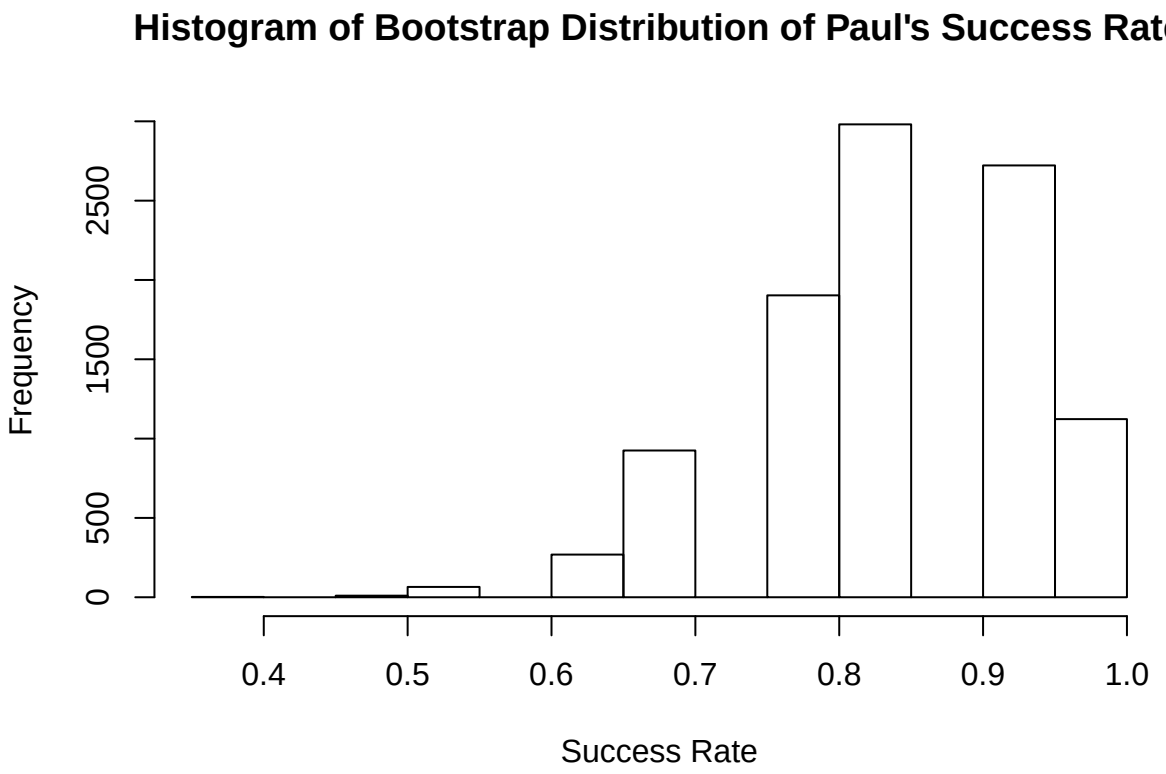
Part 2.5 (2.5 points) Make a judgement call as to whether you believe Paul is psychic based on the p-value and any other information you think is releveant. Make sure to justify your answer to explain Paul's prediction abilities.

Answer:

Our calculated p-value is less than the significance level. According to our p-value there is a 0.0108 probability that Paul could guess correctly more than 11 times, so theoretically we should reject the null hypothesis in favor of the alternative. However, with everything we know about modern day science and how the brain works, I highly doubt Paul is psychic. There are many factors that could have influenced Paul's decision, such as coloring or placement of the flags. A result like this would warrant further investigation until we can conclude that Paul is psychic.

Part 2.6 (10 points) Calculate the confidence interval for Paul's prediction abilities using the bootstrap methods

```r
# a vector of Paul's answers whether they were correct or incorrect (not necessarily in the order)
pauls_answers <- c(rep('correct', 11), 'incorrect', 'incorrect')    # 11 correct, 2 incorrect


# continue creating the bootstrap distribution from here
bootstrap_dist_paul = c()
for (i in 1:10000){
  bootstrap_sample <- sample(pauls_answers, replace = TRUE)
  bootstrap_dist_paul[i] <- sum(bootstrap_sample == "correct")/length(bootstrap_sample)
}

# plot the boostrap distribution
hist(bootstrap_dist_paul,
     main = "Histogram of Bootstrap Distribution of Paul's Success Rate",
     xlab = "Success Rate", ylab = "Frequency")
```




```
  # CI using the bootstrap based on a normal approximation to the bootstrap distribution
  (CI_paul <- paul_stat + sd(bootstrap_dist_paul)*c(-2,2))
```

```
## [1] 0.6479314 1.0443762
```

```
  # CI using the bootstrap percentile method
  (CI_perc <- quantile(bootstrap_dist_paul, c(0.025, 0.975)))
```

```
##      2.5%     97.5%
## 0.6153846 1.0000000
```

Answer:

CI using the bootstrap distribution method: [0.6479314, 1.0443762]

CI using the bootstrap percentile method: [0.6153846, 1]

Part 2.7 (5 points) There is also a formula for calculating the standard error of a proportion which is:

$s_p = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$

Use this formula to create 95% confidence intervals and compare it to the 95% confidence intervals you calculated in the previous problem. Do they appear to be similar?

```r
se_formula <- sqrt(((paul_stat)*(1 - paul_stat))/13)
(CI_formula <- paul_stat + se_formula*c(-2,2))
```

```
## [1] 0.6460173 1.0462903
```

Answer

The confidence interval calculated from the formula is [0.6460173, 1.0462903]. This appears to be most similar to the one calculated using the bootstrap distribution method, although all three are pretty similar. Only the bootstrap percentile method does not exceed 1 (indicating a 100% success rate) on the upper end.

Problem 3: Permutation tests for comparing two means - Sleep or Caffeine for Memory?

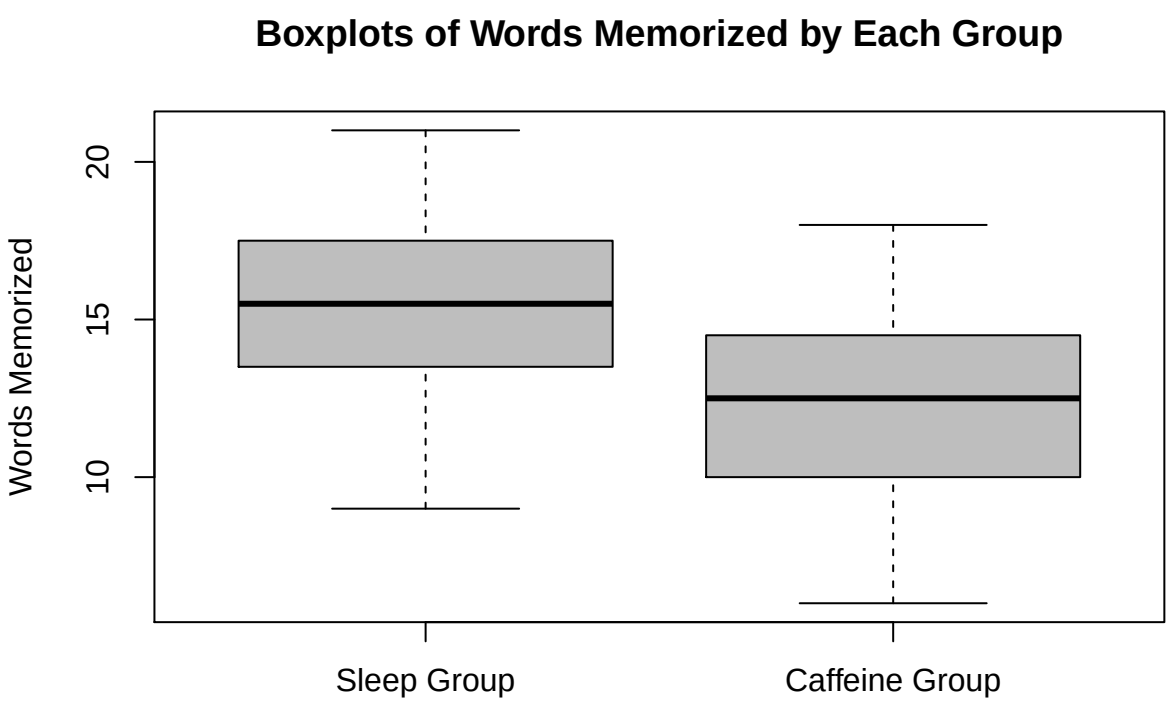
The consumption of caffeine to benefit alertness is a common activity practiced by 90% of adults in North America. Often caffeine is used in order to replace the need for sleep. One recent study compared students' ability to recall memorized information after either the consumption of caffeine or a brief sleep (see Mednick et al., 2018

A random sample of 35 adults (between the ages of 18 and 39) were randomly divided into three groups and verbally given a list of 24 words to memorize. During a break, one of the groups took a nap for an hour and a half, another group was kept awake and then given a caffeine pill an hour prior to the testing, and a third group was given a placebo. The response variable of interest is the number of words participants are able to recall following the break.

Let's run a hypothesis test to see if there is statistically significant difference in the mean number of words recalled between the group that got *sleep* and the group that got *caffeine* (we will ignore the placebo group since I don't have data from that group).

Part 3.0 (5 points) The data for the number of words recalled by members in each of the two groups is below. Start by creating a side by side boxplot comparing the two groups. Describe below whether there appears to be a difference between the groups.

```r
sleep_condition <- c(14, 18, 11, 13, 18, 17, 21, 9, 16, 17, 14, 15)
caffeine_condition <- c(12, 12, 14, 13, 6, 18, 14, 16, 10, 7, 15, 10)

boxplot(sleep_condition, caffeine_condition,
        main = "Boxplots of Words Memorized by Each Group",
        names = c("Sleep Group", "Caffeine Group"), ylab = "Words Memorized", col = "gray")
```



Answer:

From the boxplot it seems that the group that slept was able to memorize more words than the group that took caffeine. Each quartile statistic is greater than that of the caffeine group.

In parts 3.1 to 3.5 you will now do the 5 steps to run a hypothesis test.

Part 3.1 (5 points) State the null and alternative hypotheses using words and symbols. Also describe the significance level is and denote it with the appropriate symbol.

Answer:

H_0 : There is no difference in the memorization ability of people who sleep versus caffeine.

$\mu_{sleep} - \mu_{caffeine} = 0.$

H_A : There is a difference in the memorization ability of people who sleep versus caffeine

$\mu_{sleep} - \mu_{caffeine} \neq 0.$

Significance level: $\alpha = 0.05$

Part 3.2 (5 points) Calculate the value of that statistic for the observed sample, and use the appropriate symbol notation along with its value below.

```r
#Sleep group
mean(sleep_condition)
```

```
## [1] 15.25
```

```r
#Caffeine group
mean(caffeine_condition)
```

```
## [1] 12.25
```

```r
(obs_stat <- mean(sleep_condition) - mean(caffeine_condition))
```

```
## [1] 3
```

Answer:

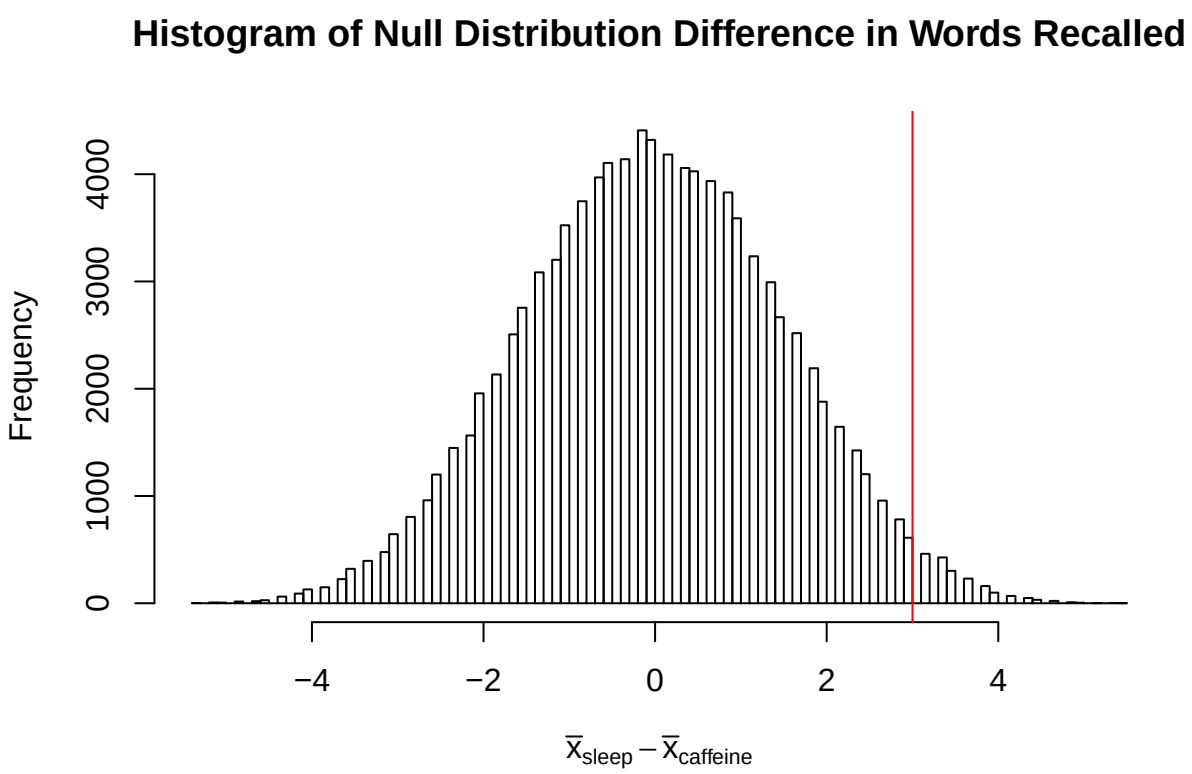
$\bar{x}_{sleep} = 15.25$

$\bar{x}_{caffeine} = 12.25$

$\bar{x}_{sleep} - \bar{x}_{caffeine} = 3$

Part 3.3 (10 points) Create a null distribution using a permutation test (i.e., combine data from both groups, randomly assign them to a fake "caffeine" and "sleep group", calculate a null statistic, and repeat 100,000 times to get a null distribution). Also plot a histogram of the null distribution and add a red vertical line to the plot at the value of the observed statistic.

```r
combined <- c(sleep_condition, caffeine_condition)
null_dist <- c()
for (i in 1:100000){
  shuff_data <- sample(combined)
  shuff_sleep <- shuff_data[1:12]
  shuff_caffeine <- shuff_data[13:24]

  null_dist[i] <- mean(shuff_sleep) - mean(shuff_caffeine)
}
hist(null_dist, main = "Histogram of Null Distribution Difference in Words Recalled",
     xlab = TeX(("$\\bar{x}_{sleep} - \\bar{x}_{caffeine}$")),
     ylab = "Frequency",
     nclass = 100)
abline(v = obs_stat, col = "red")
```



Part 3.4 (5 points) Now calculate the p-value in the R chunk below.

```
(p_val <- sum(null_dist >= obs_stat, null_dist <= -obs_stat)/100000)
```

```
## [1] 0.05049
```

Part 3.5 (2.5 points) Are the results statistically significant? What would we conclude if we used a strictly "Neyman-Pearson paradigm" where we only reject results that are less than our significance level? Do you believe there is a difference between these groups?

Answers:

The p-value is 0.05123, which is greater than our significance level $\alpha = 0.05$, so under the Neyman-Pearson paradigm the we cannot concluded that the results are statistically significant and we fail to reject the null hypothesis that there is no difference between the groups. However, I do believe there is a difference between the groups based on past scientific evidence which consistently confirms the importance of sleep for memory retention, as well as the fact that the p-value is very close to α. The boxplots do show a difference in the values as well. If this test was a one-tailed test attempting to find whether $\bar{x}_{sleep} > \bar{x}_{caffeine}$, we would find that the p-value is smaller than α.

Part 3.6 (5 points) Parametric hypothesis tests are hypothesis tests where the null distribution is given by a mathematical density function. When comparing two means, a parametric hypothesis test, that you likely learned about in introductory statistics, is the t-test, where the null distribution is a t-distribution.

R has a built in function called `t.test(sample1, sample2)` that takes two samples of data and runs a t-test on them. Use this function to compare the sleep and caffeine groups and report if there is a statistically significant difference between the groups. Also report the 95% confidence interval that the t.test function returns and describe whether this confidence interval is consistent with it being plausiable that there is no difference between the population means of these two groups.

```
t.test(sleep_condition, caffeine_condition)
```

```
##
##  Welch Two Sample t-test
##
## data:  sleep_condition and caffeine_condition
## t = 2.1438, df = 21.894, p-value = 0.04342
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.09699633 5.90300367
## sample estimates:
## mean of x mean of y
##     15.25     12.25
```

Answers:

The p-value is 0.04342, which is less than $\alpha = 0.05$, so we can conclude that the results are statistically significant and there is evidence that there is a difference between the groups. The 95% confidence interval is [0.097, 5.903]. Since this interval only includes positive values and no 0 value, it not consistent with it being plausible that there is no different in the means of the two groups. Instead, this interval is consistent with the hypothesis that words memorized by the sleep group are great than the number of words memorized by the caffeine group.

Reflection (5 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 4

Homework 5

The purpose of this homework is to practice using randomization methods to run hypothesis tests for more than two means and for correlation. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday October 6th.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Part 1: Hypothesis tests for more than two means

Are movies that have particular Motion Picture Association of America (MPAA) ratings (i.e., G, PG, PG-13 or R) enjoyed more by movie critics? In the exercises below we will run hypothesis tests to examine this question using data from ~456 movies randomly selected from the Rotten Tomatoes website.

The code below loads the movie data in an object called `movies` and it also creates an object called `movies3` which only keeps movies with ratings of G, PG, PG-13 and R. For all the exercises below, **only use the data in the movies3 data frame**. For a codebook describing the variables in this dataframe, please see this website.

```
# load the data
load('movies.Rdata')

# only keep movies rated "G", "PG", "PG-13", "R"
movies3 <- movies[movies$mpaa_rating %in% c("G", "PG", "PG-13", "R"), ]
movies3$mpaa_rating <- droplevels(movies3$mpaa_rating)
```

Part 1.1 (5 points) - step 0 Let's start our analysis by describing and plotting the data. Please report the number of cases and the number of variables in the `movies3` data frame, and what each case corresponds to. Also, create a side-by-side boxplot comparing the critics' scores of the movie for each MPAA rating level. Does it appear that the critics' scores differ on average depending on the MPAA classification of the movie?

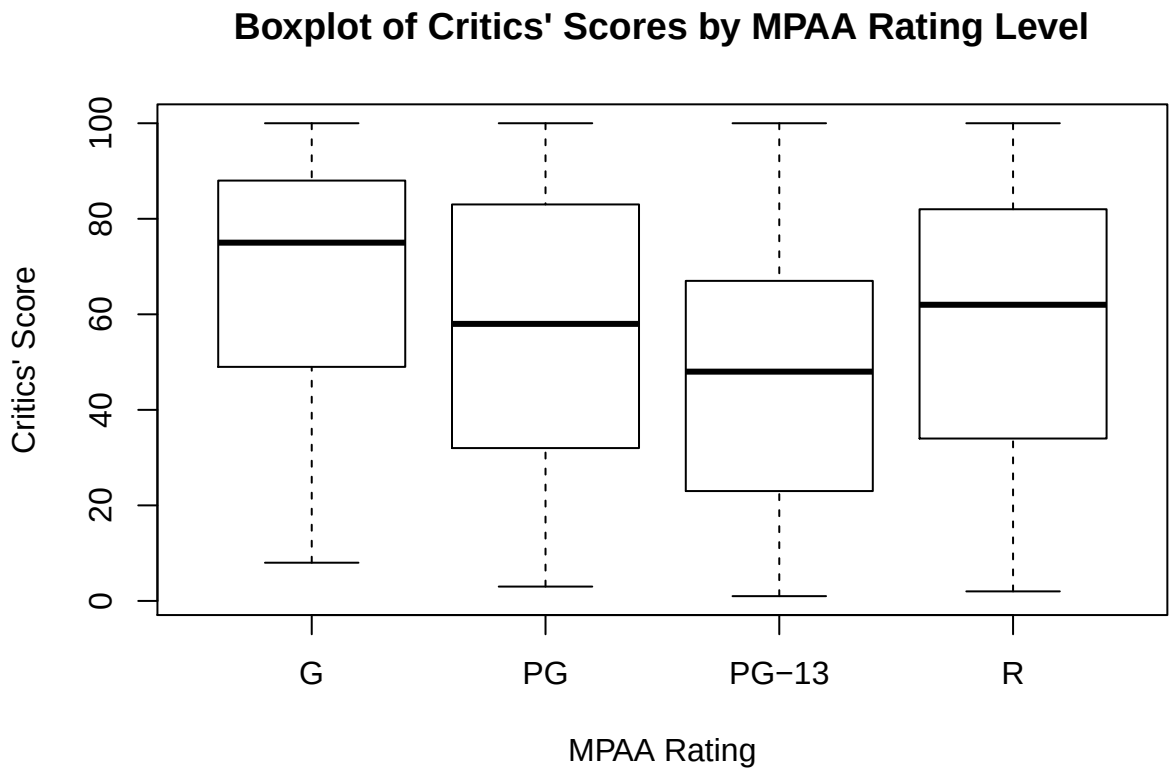
```
summary(movies3)
```

```
##     title                title_type                genre
##  Length:599         Documentary : 23   Drama            :293
##  Class :character   Feature Film:573   Comedy           : 87
##  Mode  :character   TV Movie    :  3   Action & Adventure: 65
##                                        Mystery & Suspense: 59
##                                        Documentary      : 21
##                                        Horror           : 21
##                                        (Other)          : 53
```

```
##     runtime    mpaa_rating                              studio
##  Min.   : 40   G    : 19   Paramount Pictures              : 37
##  1st Qu.: 93   PG   :118   Warner Bros. Pictures           : 30
##  Median :103   PG-13:133   Sony Pictures Home Entertainment: 27
##  Mean   :106   R    :329   Universal Pictures              : 23
##  3rd Qu.:116               Warner Home Video               : 19
##  Max.   :202               (Other)                         :456
##                            NA's                            :  7
##  thtr_rel_year  thtr_rel_month    thtr_rel_day    dvd_rel_year
##  Min.   :1970   Min.   : 1.000   Min.   : 1.00   Min.   :1991
##  1st Qu.:1990   1st Qu.: 4.000   1st Qu.: 7.00   1st Qu.:2001
##  Median :1999   Median : 7.000   Median :15.00   Median :2003
##  Mean   :1997   Mean   : 6.783   Mean   :14.53   Mean   :2004
##  3rd Qu.:2006   3rd Qu.:10.000   3rd Qu.:21.50   3rd Qu.:2007
##  Max.   :2014   Max.   :12.000   Max.   :31.00   Max.   :2015
##                                                  NA's   :7
##  dvd_rel_month     dvd_rel_day     imdb_rating    imdb_num_votes
##  Min.   : 1.000   Min.   : 1.00   Min.   :1.900   Min.   :   390
##  1st Qu.: 3.000   1st Qu.: 7.00   1st Qu.:5.900   1st Qu.:  5576
##  Median : 6.000   Median :15.00   Median :6.500   Median : 17190
##  Mean   : 6.299   Mean   :14.96   Mean   :6.418   Mean   : 62018
##  3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:7.200   3rd Qu.: 64681
##  Max.   :12.000   Max.   :31.00   Max.   :9.000   Max.   :893008
##  NA's   :7        NA's   :7
##         critics_rating critics_score    audience_rating audience_score
##  Certified Fresh:118    Min.   :  1.00   Spilled:270     Min.   :11.00
##  Fresh          :178    1st Qu.: 31.00   Upright:329     1st Qu.:45.00
##  Rotten         :303    Median : 59.00                   Median :63.00
##                         Mean   : 55.41                   Mean   :61.01
##                         3rd Qu.: 80.00                   3rd Qu.:78.50
##                         Max.   :100.00                   Max.   :97.00
##
##  best_pic_nom best_pic_win best_actor_win best_actress_win best_dir_win
##  no :577      no :592      no :509        no :529          no :556
##  yes: 22      yes:  7      yes: 90        yes: 70          yes: 43
##
##
##
##
##
##  top200_box   director            actor1             actor2
##  no :584    Length:599         Length:599         Length:599
##  yes: 15    Class :character   Class :character   Class :character
##             Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     actor3             actor4             actor5
##  Length:599         Length:599         Length:599
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
```

```
##
##
##    imdb_url            rt_url
##  Length:599         Length:599
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

```
boxplot(movies3$critics_score ~ movies3$mpaa_rating,
        main = "Boxplot of Critics' Scores by MPAA Rating Level",
        xlab = "MPAA Rating", ylab = "Critics' Score")
```



Answers:

There are 599 cases (corresponding to 599 total movie titles), and 32 variables observed. It does appear that the scores from critics changes depending on the MPAA Rating: The scores for PG-13 movies seem to be lower on average, while the scores for G-rated movies seem to be higher.

Part 1.2 (5 points): Let's examine whether there is a statistically significant difference in the mean critics' scores for each MPAA level. Start by stating the null and alternative hypotheses in symbols and words, and also state the alpha level that is most commonly used.

In words

Null Hypothesis: There is no difference in the mean critics' scores for each MPAA level.

Alternative Hypothesis: There is a statistically significant difference in the mean critics' scores for each MPAA level (there is at least one difference)

In symbols

$H_0 : \mu_G = \mu_{PG} = \mu_{PG-13} = \mu_R$

$H_A : \mu_i \neq \mu_j$ for some i, j

The significance level

$\alpha = 0.05$

Part 1.3 (5 points): For our first analyses, let's use the MAD statistic that we discussed in class to compare the mean critic scores between the different MPAA rating levels. Do the following steps:

1) Extract a vector from the movies3 data frame that has the critics' scores and store it in an object called `critic_scores`
2) Extract a vector from the movies3 data frame that has the MPAA ratings and store it in an object called `MPAA_ratings`
3) Call the `get_group_means(data, grouping)` function I wrote above to get the mean of the critics' scores for each MPAA rating. Save these means in an object called `group_means`
4) Call the `get_MAD_stat(group_means)` function I wrote above to get the MAD statistic.

Report the group means values below along with the MAD statistic value.

```
# store the critics scores and the MPAA ratings in objects
critic_scores <- movies3$critics_score
MPAA_ratings <- movies3$mpaa_rating

# get the mean critics's scores for each MPAA rating level
# Group means shown below for G, PG, PG-13, R respectively
(group_means <- get_group_means(critic_scores, MPAA_ratings))
```

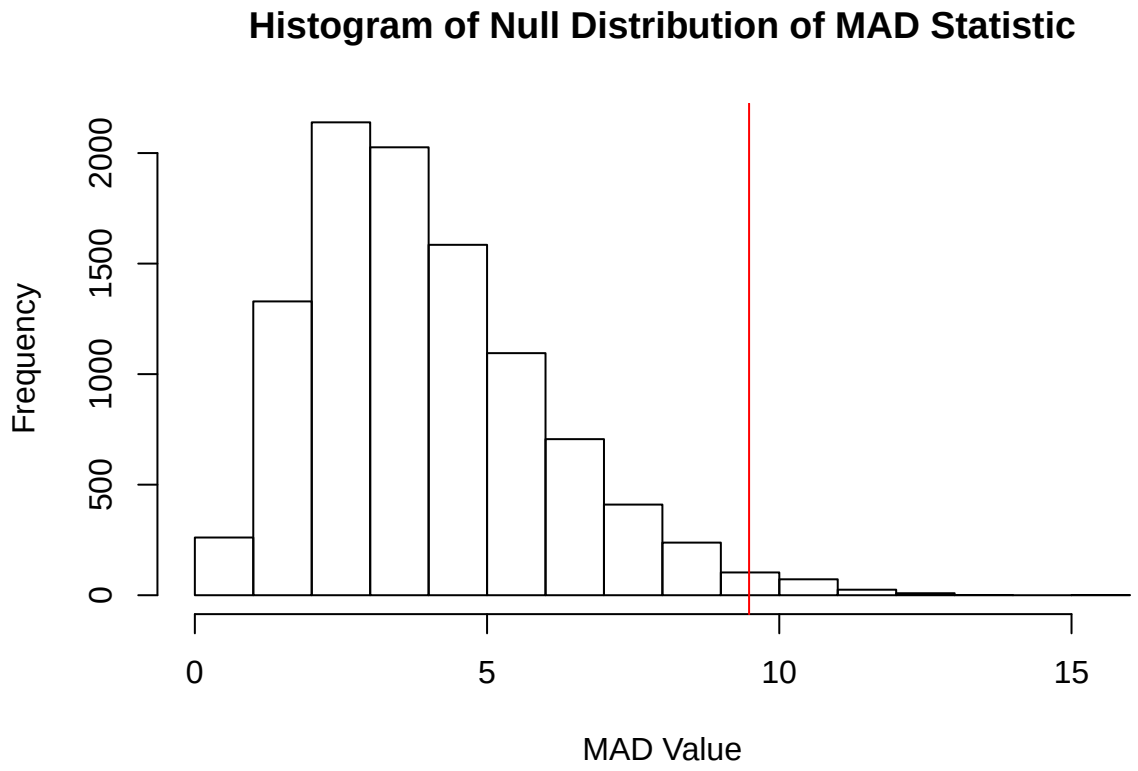
```
## [1] 65.94737 56.79661 47.24812 57.59878
```

```
# Calculate the MAD statistic
(obs_stat <- get_MAD_stat(group_means))
```

```
## [1] 9.48332
```

Part 1.4 (10 points): Now run steps 2-5 of the hypothesis test, as discussed in class. Be sure to plot the null distribution along with a red vertical line at the real MAD statistic value. Also report the p-value below. Based on this analysis comparing group means using the MAD statistic, does there appear to be a difference between between the critic's scores depending on the MPAA rating? (note: please use the answer section below to answer any questions that are posed in this and future homeworks).

```r
# create the null distribution
null_dist <- NULL
for (i in 1:10000){

  shuff_rating <- sample(movies3$mpaa_rating)
  shuff_group_means <- as.vector(by(movies3$critics_score, shuff_rating, mean))
  null_dist[i] <- get_MAD_stat(shuff_group_means)

}


# plot the null distribution with a red vertical line for the statistic value

hist(null_dist, main = "Histogram of Null Distribution of MAD Statistic",
     xlab = "MAD Value", ylab = "Frequency")
abline(v = obs_stat, col = "red")
```




```r
# report the p-value

(p_val <- sum(null_dist >= obs_stat)/length(null_dist))
```

```
## [1] 0.0158
```

Answers

The p-value is 0.0158. Since this is less than our significance level $\alpha = 0.05$, there is statistically significant evidence that there is difference in the critics' scores based on MPAA rating.

Part 1.5 (10 points): We could also run the hypothesis test comparing the means based on using an F-statistic. The equation for an F-statistic is:

$$F = \frac{\text{between-group variability}}{\text{within-group variability}} = \frac{\frac{1}{K-1}\sum_{i=1}^{K} n_i(\bar{x}_i - \bar{x}_{tot})^2}{\frac{1}{N-K}\sum_{i=1}^{K}\sum_{j=1}^{n_i}(x_{ij} - \bar{x}_i)^2}$$

In the above equation, the symbols mean the following:

- K = 4 corresponds to the 4 MPAA rating levels (G, PG, PG-13, and R)
- N corresponds to the total number of movies we are using in our analysis
- x_{ij} corresponds to the j^{th} movie with a rating in the i^{th} MPAA rating level
- n_i is the number of movies in the i^{th} group (e.g., the number of movies with a rating of G)
- $\bar{x}_i$ is the average score for the i^{th} group (e.g., the average score for movies with a rating of G)

We will discuss this equation a bit more when we discuss ANOVAs, but for now we can just think of it as a number that describes differences in the means of our data.

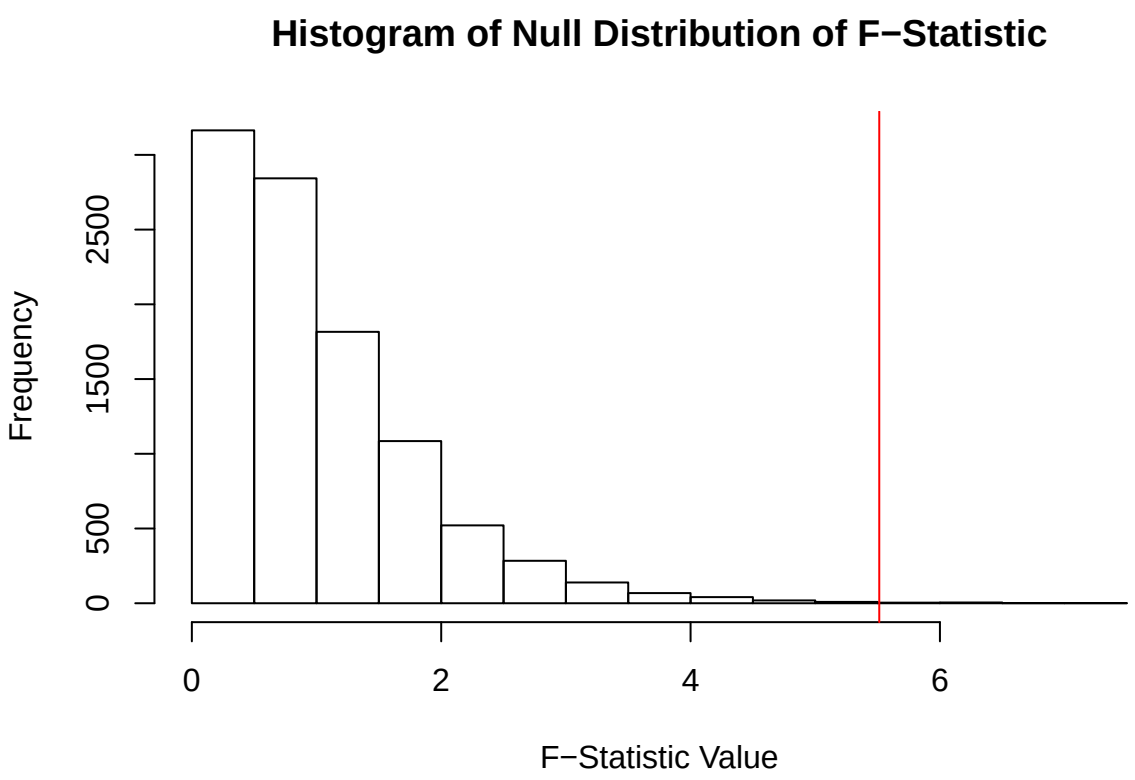
To calculate the F-statistic, I have written a function called `get_F_statistic(data, grouping)`. This function takes a vector of data values, and a vector indicating which group each data value belongs to.

Please rerun steps 2-5 of a hypothesis test (i.e., permutation test) using F-statistic in the R chunk below. Again be sure to plot the null distribution along with a red vertical line a the real observed F-statistic value. Also report the p-value below. Based on this analysis comparing group means using the F-statistic, does there appear to be a difference between between the critic's scores depending on the MPAA rating?

```
# calculate the observed statistic using the get_F_statistic function
(obs_stat_F <- get_F_statistic(critic_scores, MPAA_ratings))
```

```
## [1] 5.513491
```

```
# create the null distribution
null_dist_F <- NULL
for (i in 1:10000){

  shuff_rating <- sample(movies3$mpaa_rating)
  null_dist_F[i] <- get_F_statistic(movies3$critics_score, shuff_rating)

}


# plot the null distribution with a red vertical line for the statistic value
hist(null_dist_F, main = "Histogram of Null Distribution of F-Statistic",
     xlab = "F-Statistic Value", ylab = "Frequency")
abline(v = obs_stat_F, col = "red")
```




```
# report the p-value

(p_val_F <- sum(null_dist_F >= obs_stat_F)/length(null_dist_F))
```

```
## [1] 0.0011
```

Answers

The p-value is 0.0011, which means we should reject our null hypothesis. There appears to be a difference in critics' scores based on rating.

Part 1.6 (10 points): Let's try running a permutation test using one more statistic, namely the the statistic that returns $max\ \bar{x}_i - \bar{x}_j$, where $\bar{x}_i$, and $\bar{x}_j$ refer to mean critic score for the i^{th} and j^{th} movie rating levels. To do this analysis, start by writing a function yourself called 'get_max_diff(data, grouping)' that takes a data vector (i.e., the critics ratings) and a grouping vector (i.e., the MPAA ratings), and returns the maximum value for the difference between the means over all pairs of groups.

Hints: One way you can write this by modifying the `get_MAD_stat` function (e.g., start by copying and pasting it to the chunk below). Again use two nested for loops, but instead of summing the results, use an if statement and to store the difference between mean scores if the current difference is greater than any difference seen on previous iterations (i.e., create an object called `max_diff` that initially has a value of 0, and update this value when in any interation of the loop that has a greater value for the difference between means).

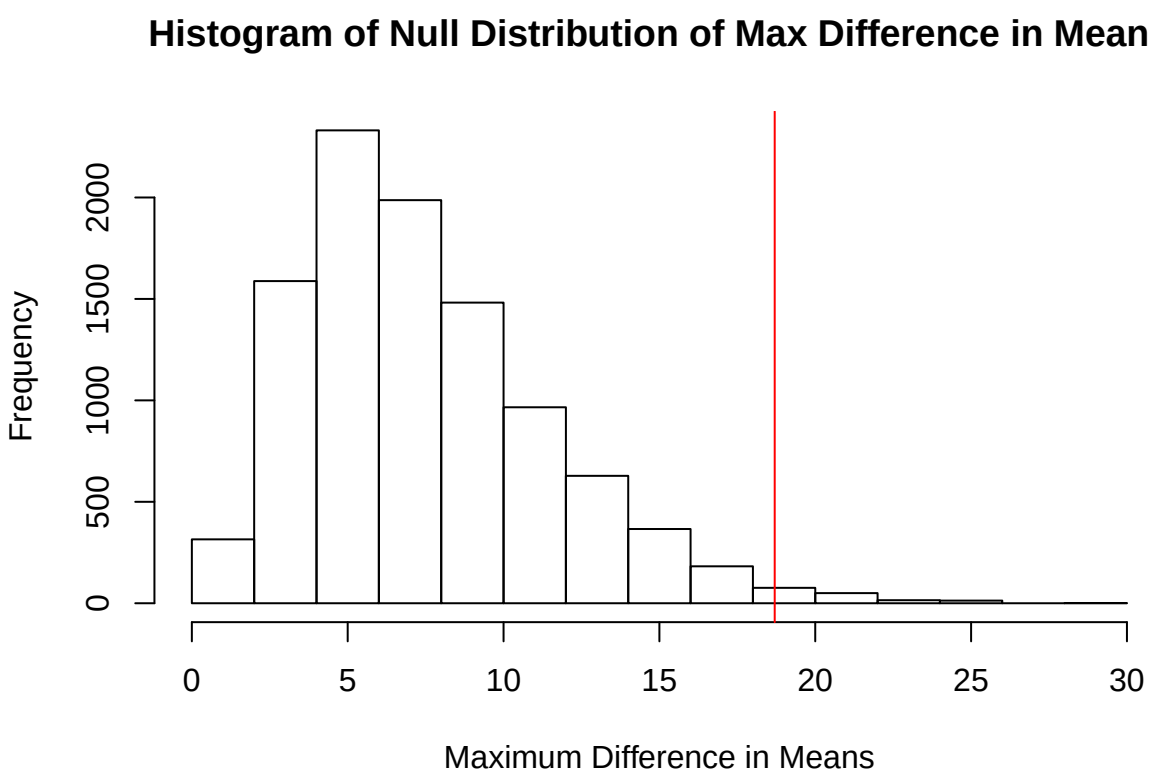
Once you have written this function, calculate and report the observed statistic value using this function.

```r
# a function to the maximum absolute difference between all pairs of means
get_max_diff <- function(data, grouping) {
  max_diff <- 0
  means_vec <- as.vector(by(data, grouping, mean))

  for (iGroup1 in 1:(length(means_vec) - 1)) {

    for (iGroup2 in (iGroup1 + 1):(length(means_vec))){
      diff <- abs(means_vec[iGroup1] - means_vec[iGroup2])
      if(diff > max_diff){
        max_diff <- diff
      }
    }
  }
    max_diff
}

# end of the function


# use the function to get the observed statistic

(obs_stat_max <- get_max_diff(critic_scores, MPAA_ratings))
```

```
## [1] 18.69925
```

Part 1.7 (10 points): Now Repeat steps 2-5 of hypothesis testing using 'get_max_diff(data, grouping)' function. Again be sure to plot the null distribution along with a red vertical line a the real observed max-diff statistic value, report the p-value below, and answer the question about whether based on this analysis comparing group means using the max-diff statistic, does there appear to be a difference between between the critic's scores depending on the MPAA rating?

```r
# create the null distribution
null_dist_max <- NULL
for (i in 1:10000){
  shuff_ratings <- sample(movies3$mpaa_rating)
  null_dist_max[i] <- get_max_diff(movies3$critics_score, shuff_ratings)
}

# plot the null distribution with a red vertical line for the statistic value

hist(null_dist_max, main = "Histogram of Null Distribution of Max Difference in Means",
     xlab = "Maximum Difference in Means", ylab = "Frequency" )
abline(v = obs_stat_max, col = "red")
```




```
# report the p-value
(p_val_max <- sum(null_dist_max >= obs_stat_max)/length(null_dist_max))
```

```
## [1] 0.0118
```

Answers

The p-value is 0.0118, which is less than significance level $\alpha = 0.05$. Based on this analysis there does seem to be a difference in critic scores based on MPAA rating.

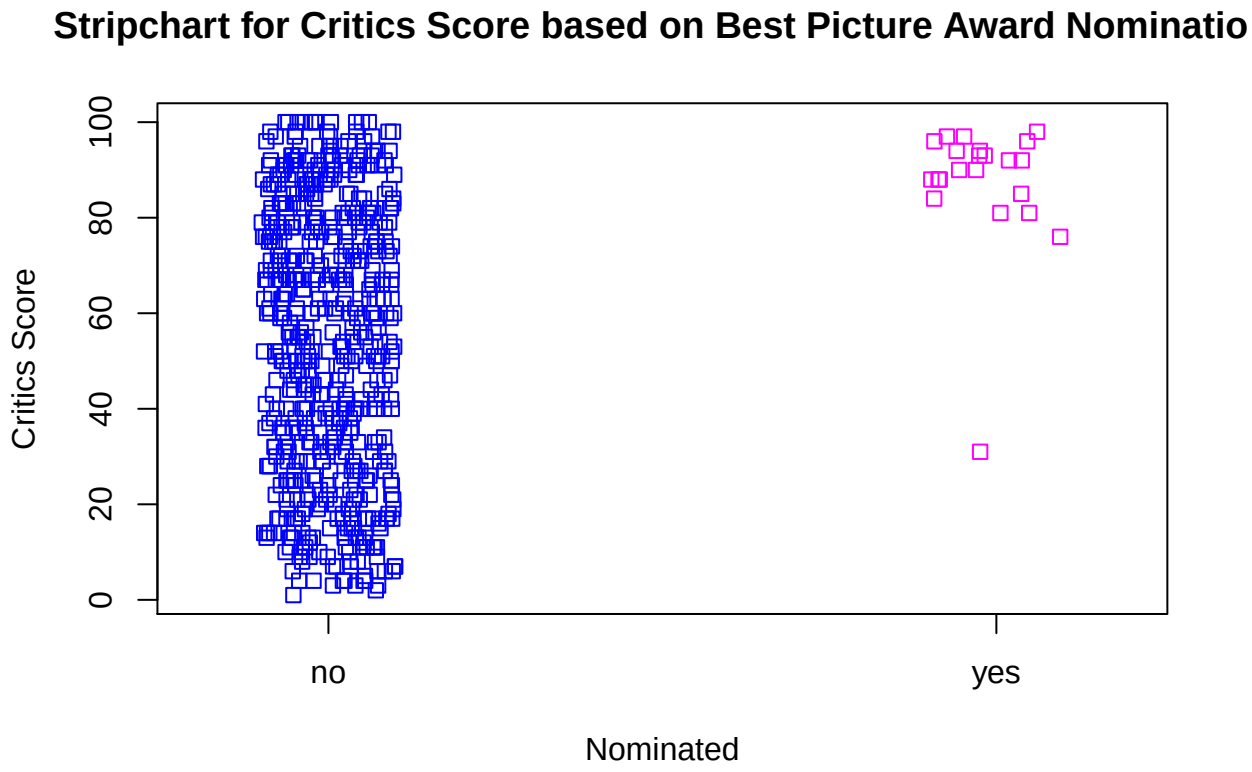
Part 1.8 (5 points): In the three exercises above, you ran three permutation tests based on three different statsitics. Describe which statistic/permutation test seems best and why?

Answers

From these three permutation tests it seems that the F-statistic test is best. With the maximum difference test, it seems that the variance in means could easily be overestimated by outlier statistics, since we are only looking at the difference between the means of two groups instead of all of them. With the Mean Absolute Difference, this difference is slightly diminished because we are considering the difference between the means of each group. However, the F-statistic seems to be the most comprehensive because it takes into account both the variability between the group means and variability within the group. From the p-values we can also see that the F-statistic test yielded the strongest result (lowest p-value), which also suggests strength as a test.

Part 1.9 (5 points): Using the R chunk below, explore the movies data more and create one additional plot that shows something interesting.

```
#tbl_df(movies3) #just to visualize a table of the movies data, not shown

#I created a stripchart comparing the scores from critics
#based on whether they received a "Best Picture" nomination
stripchart(movies3$critics_score ~ movies3$best_pic_nom,
           vertical = T, col = c("blue", "magenta"), method = "jitter",
           main = "Stripchart for Critics Score based on Best Picture Award Nomination",
           xlab = "Nominated",
           ylab = "Critics Score")
```



Problem 2: Permutation tests for correlation

Part 1: The 1969 draft lottery

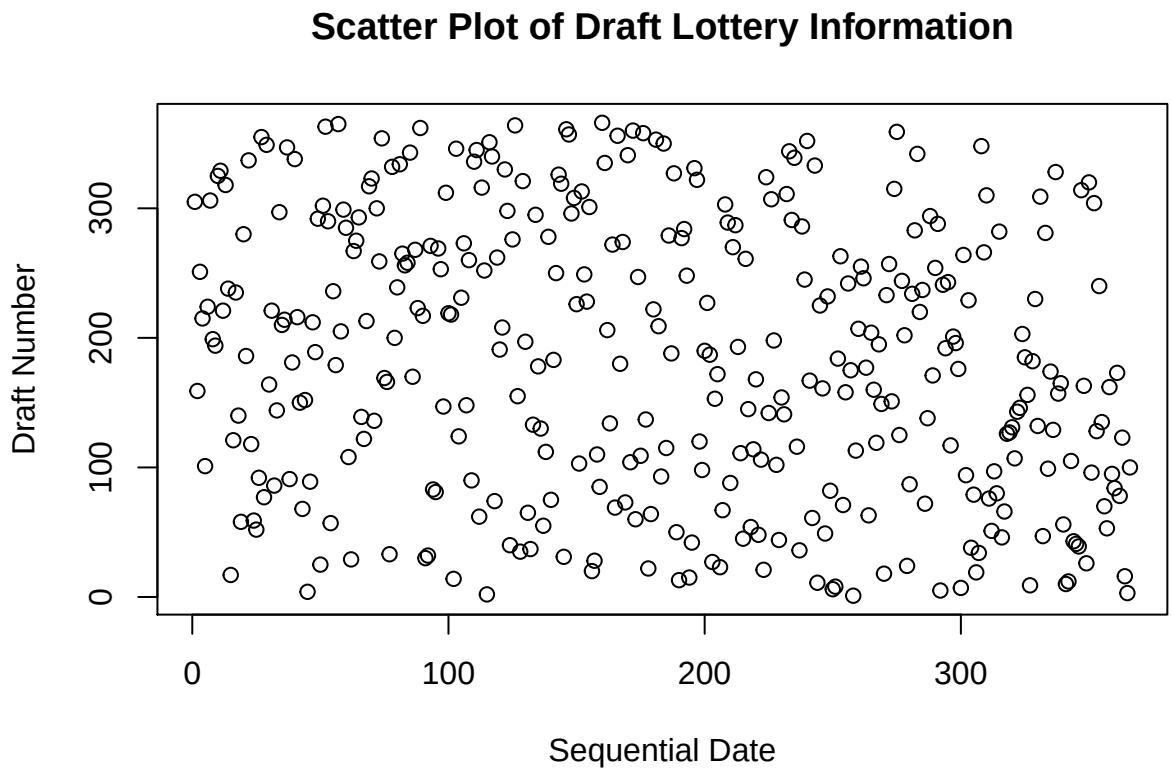
In 1969, the United States Selective Service conducted a lottery to decide which young men would be drafted into the armed forces. Each of the 366 birthdays in a year (including February 29) was assigned a draft number. Young men born on days that were assigned low draft numbers were drafted.

If the draft was completely fair, there should be no correlation between the draft number and the date

someone was born. In the following set of exercises we will use hypothesis testing to assess whether there was indeed no correlation.

Part 2.0 (2 points): Let's start our analyses, as usual, by visualizing the data. A data frame that contains the draft lottery information can be loaded into R using the code below. This data frame contains two variables (columns). The first column contains sequential days of the year and the second column contains the draft number associated with that date. Create a scatter plot of the draft number as a function of the sequential date. Does there appear to be any trend in the data?

```
# load the data into R
load('draft_lottery_data.Rda')

# plot the data
plot(draft_lottery_data$Draft_Number ~ draft_lottery_data$Sequential_Date,
     main = "Scatter Plot of Draft Lottery Information",
     xlab = "Sequential Date",
     ylab = "Draft Number")
```



Answer:

Looking at the scatter plot there does not appear to be any trend in the data.

Part 2.1 (3 points): Now let's do step 1 of our null hypothesis significance tests (NHSTs) by stating the null and alternative hypotheses in symbols and in words, and state the significance level.

Answer:

In Words:

Null hypothesis: There is no correlation between the date picked and the draft number assigned.

Alternative hypothesis: There is a correlation between the date picked and the draft number assigned.

In Symbols:

$\mu_0 : \rho = 0$

$\mu_A : \rho \neq 0$

Part 2.2 (2 points): Next let's do step 2 of hypothesis testing by calculating the statistic of interest and save it to a variable obs_stat. Describe what this statistic means as clearly as you can (e.g., if the statistic is negative what does that mean in terms of dates and draft numbers?).

```
(obs_stat <- cor(draft_lottery_data$Draft_Number,
                 draft_lottery_data$Sequential_Date))
```

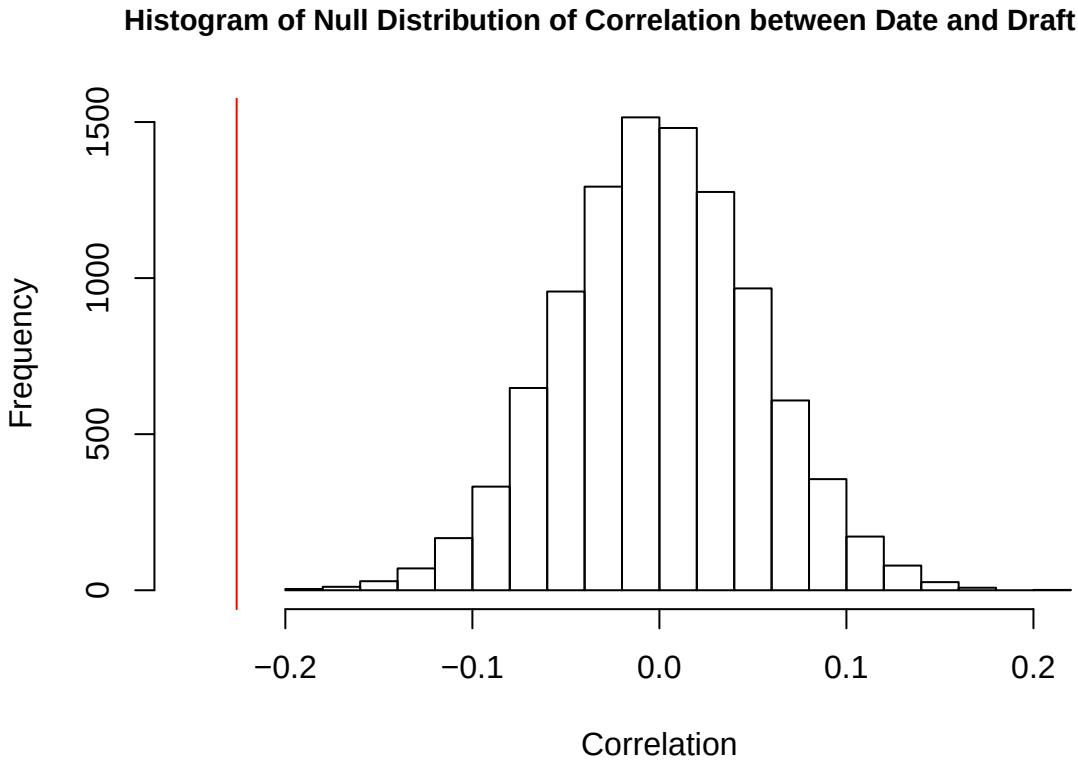
```
## [1] -0.2260414
```

Answer:

The observed correlation is -0.2260414. This means there is a moderate negative correlation between Draft Number and Date of Birth, so the later the sequential date, the lower the draft number was likely to be.

Part 2.3 (5 points): Now let's do step 3 of hypothesis testing by creating a null distribution. You can calculate one point in the null distribution by shuffling one of the variables (using the `sample()` function) and then calculating the correlation. Use a for loop to repeat this process 10,000 times to generate the full null distribution. Plot a histogram of the null distribution, put a red veritical line on it at the value of the observed statistic, and describe the null distributions shape. Is the center of the null distribution at a value that makes sense to you?

```
# create the null distribution and plot it
null_dist <- NULL
for (i in 1:10000){
  shuff_data <- sample(draft_lottery_data$Sequential_Date)
  null_dist[i] <- cor(draft_lottery_data$Draft_Number, shuff_data)
}

hist(null_dist,
     main = "Histogram of Null Distribution of Correlation between Date and Draft Number",
     xlab = "Correlation", ylab = "Frequency", xlim = c(-0.25, 0.25), cex.main = 0.9)
abline (v = obs_stat, col = "red")
```



Answer:

The shape of the null distribution is approximately normal. The center is at approximately 0, which makes sense because if there is no correlation then $\rho = 0$. This is acheived by shuffling (randomizing) the data to be used in calculating correlation, so there should be the most calculated correlations close to 0.

Part 2.4 (5 points): Now use the vector null_dist and the obs_stat to calculate the p-value by seeing the proportion of points in the null distribution that are *as extreme or more extreme* than the observed statistic. Is this p-value consistent with there being no correlation between draft numbers and sequential dates?

```
(p_val <- sum(abs(null_dist) >= abs(obs_stat))/length(null_dist))
```

```
## [1] 0
```

Answer:

The p-value is equal to 0, thus we should reject the null hypothesis. There is evidence that there is some correlation between date and draft number.

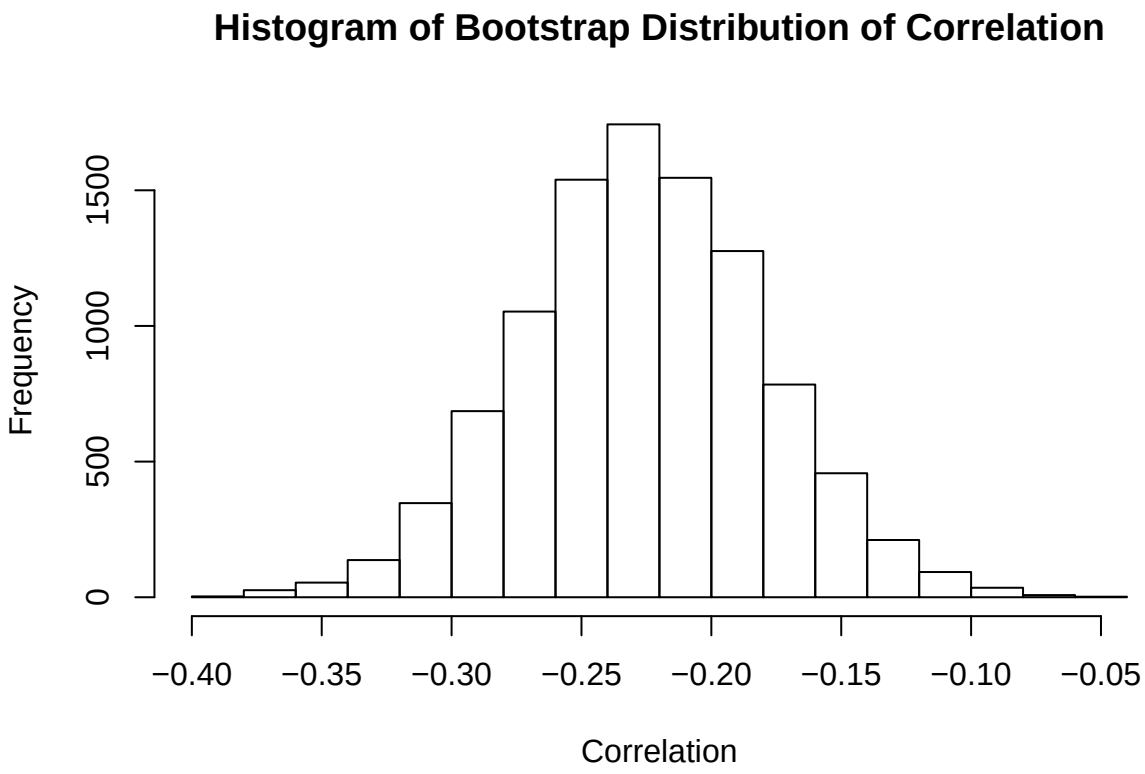
Part 2.5 (5 points): Make a judgement call as to whether you believe the draft lottery was fair. Make sure to justify your answer.

Answer:

I don't believe the draft lottery was fair because according to our analysis, the probability of such results occuring (that the correlation is -0.22) is zero. Since the numbers were physically drawn from slips of paper (Source: Wikepedia) it is possible that the papers were put into a box in a certain order and not mixed well enough to acheive a random distribution.

Part 2.6 (10 points): Calculate a 95% confidence interval for the value of the correlation between sequential date and draft number using the bootstrap. Note that you can sample points in a *data frame* with replacement using: bootstrap_data_frame <- sample(draft_lottery_data, size = 366, replace = TRUE). Does the confidence interval contain 0, and would you expect it to contain 0?

```
 # an example of a bootstrapped data frame
 one_bootstrap_data_frame <- draft_lottery_data[sample(1:366, 366, replace = TRUE), ]



 # Use a for loop to create a full bootstrap distribution
bootstrap_dist <- NULL
for (i in 1:10000){
 one_bootstrap_data_frame <- draft_lottery_data[sample(1:366, 366, replace = TRUE), ]
  bootstrap_dist[i] <- cor(one_bootstrap_data_frame$Sequential_Date,
                           one_bootstrap_data_frame$Draft_Number)
}



 # plot the bootstrap distribution
 hist(bootstrap_dist, main = "Histogram of Bootstrap Distribution of Correlation",
      xlab = "Correlation", ylab = "Frequency")
```




```
# create confidence intervals based on the bootstrap
(CI <- obs_stat + sd(bootstrap_dist)*c(-2,2))
```

```
## [1] -0.3200475 -0.1320353
```

Answer:

The confidence interval here is [-0.3200475, -0.1320353]. It does not contain zero, and we do not expect it to since the confidence interval should contain the true population parameter 95% of the time, and we know it is extremely unlikely that 0 is the population parameter (this would imply no correlation between Date and Draft Number).

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 5

Homework 6

The purpose of this homework is to practice running parametric hypothesis tests and simple linear regression. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday October 13th.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Part 1: Parametric tests comparing two means - the t-test

Sleep or Caffeine for Memory revisited

On problem 3 of homework 4, you used a permutation tests for comparing two means. The description on the homework of that study was:

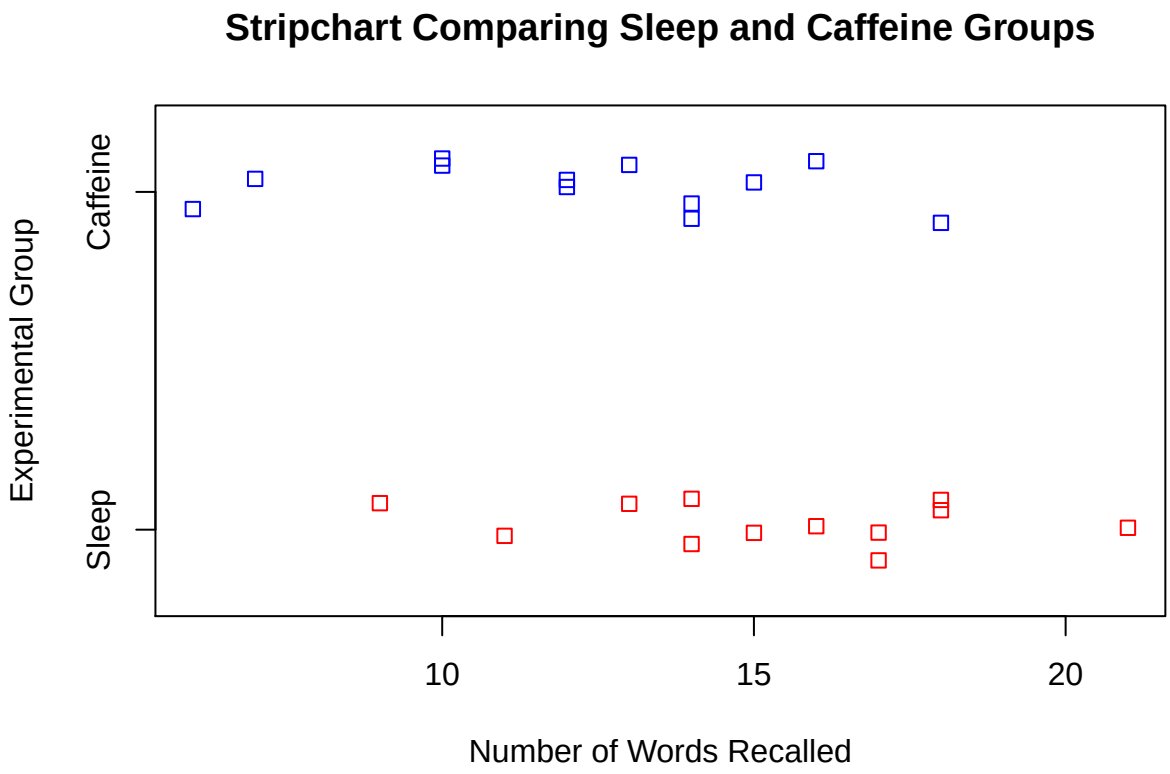
The consumption of caffeine to benefit alertness is a common activity practiced by 90% of adults in North America. Often caffeine is used in order to replace the need for sleep. One recent study compared students ability to recall memorized information after either the consumption of caffeine or a brief sleep (see Mednick et al., 2018

A random sample of 35 adults (between the ages of 18 and 39) were randomly divided into three groups and verbally given a list of 24 words to memorize. During a break, one of the groups took a nap for an hour and a half, another group was kept awake and then given a caffeine pill an hour prior to the testing, and a third group was given a placebo. The response variable of interest is the number of words participants are able to recall following the break.

Let's now run a parametric hypothesis test (t-test) to see if there is statistically significant difference in the mean number of words recalled between the group that got *sleep* and the group that got *caffeine*.

Part 1.0 (5 points) The data for the number of words recalled by members in each of the two groups is below. Start by creating a stripchart comparing the two groups. From this plot, does there appear there would be any major concerns running a t-test to analyze whether there are differences between the means of these groups?

```
sleep_condition <- c(14, 18, 11, 13, 18, 17, 21, 9, 16, 17, 14, 15)
caffeine_condition <- c(12, 12, 14, 13, 6, 18, 14, 16, 10, 7, 15, 10)

stripchart(list(sleep_condition, caffeine_condition),
           group.names = c("Sleep", "Caffeine"),
           ylab = "Experimental Group",
           xlab = "Number of Words Recalled",
           col = c("red", "blue"),
           method = "jitter",
           main = "Stripchart Comparing Sleep and Caffeine Groups")
```



Answer

Since the sample size for both populations is small (n = 12) for both, there are some concerns with using a t-test. The sample should either be large or normally distributed, but from the stripchart it appears more uniformly distributed than normally distributed.

In parts 1.1 to 1.5 you will now do the 5 steps to run a hypothesis test using parametric methods.

Part 1.1 (2 points) State the null and alternative hypotheses using words and symbols. Also describe the significance level is and denote it with the appropriate symbol.

Answer:

H_0 : There is no difference in the memorization ability of people who sleep versus people who consume caffeine.

$\mu_{sleep} - \mu_{caffeine} = 0.$

H_A : There is a difference in the memorization ability of people who sleep versus people who consume caffeine.

$\mu_{sleep} - \mu_{caffeine} \neq 0.$

Significance level: $\alpha = 0.05$. The significance level is the probability that we reject the null hypothesis when it is true (we conclude there is a difference in memorization ability when there is not). This is a Type I error.

Part 1.2 (8 points) Calculate a t-statistic for the observed sample and report the value. Based just at looking at the statistic value, does it seem that the results will end up being statistically significant? (hint: if this was a z value from a standard normal distribution, would it be statistically signficiant?).

```
(total_SE <- sqrt(var(sleep_condition)/length(sleep_condition)+
                    var(caffeine_condition)/length(caffeine_condition)))
```

```
## [1] 1.399405
```

```
(obs_tstat <- (mean(sleep_condition) - mean(caffeine_condition))/total_SE)
```

```
## [1] 2.143769
```

Answer

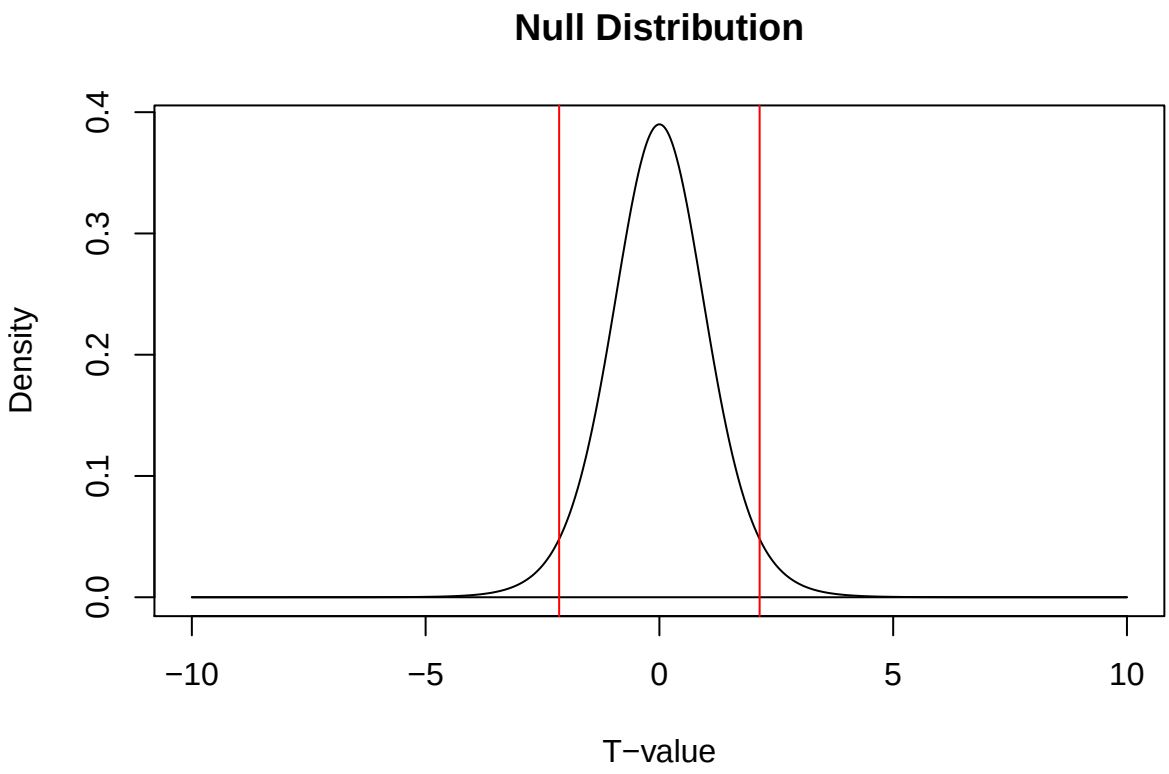
Based on the t-value it does seem that the results will be statistically significant. Since we are performing a two-tailed test, and the t-statistic is more than 2, then the density under the curve for both tails should be less than 0.05 (since two standard deviations on a standard normal distribution covers 95% of the curve). Thus p-value will be less than $\alpha = .05$ and the results will be significant.

Part 1.3 (10 points) Identify and plot the null distribution, and report the degrees of freedom. Also add red vertical lines to your plot at the observed statistic value(s) to indicate the amount of probability area in the tails of the null distribution that are more extreme than the observed statistic value.

```
# sample sizes for the two conditions
n1 <- length(sleep_condition)
n2 <- length(caffeine_condition)

# estimate the degrees of freedom
(df <- min(n1 - 1, n2 - 1))
```

```
## [1] 11
```

```
# plot the null distribution with the observed statistic on it
x_vals <- seq(-10, 10, length.out = 1000)
y_vals <- dt(x_vals, df)

plot(x_vals, y_vals, type = "l",
     main = "Null Distribution",
     xlab = "T-value",
     ylab = "Density")
points(x_vals, rep(0, length(x_vals)), type = 'l')
abline(v = obs_tstat, col = "red")
abline(v = -obs_tstat, col = "red")
```



Answer

The degrees of freedom is 11.

Part 1.4 (7 points) Now calculate the p-value in the R chunk below.

```
(p_val <- pt(obs_tstat, df = df, lower.tail = FALSE) + pt(-obs_tstat, df = df))
```

```
## [1] 0.05524225
```

Part 1.5 (15 points) As discussed in class, Null Hypothesis Signifiance Testing is a hybrid of two different theories, namely Fisher's "significance testing" and Neyman and Pearson's "hypothesis testing" (to read more about this see this paper.

Please answer the following questions to interpret the results in light of these theories:

Based on **Neyman and Pearson's "hypothesis testing" paradigm**: 1) Are the results statistical significant at a significance level of $\alpha = 0.05$? 2) Does it seem likely you are making a Type I error here? 3) Does it seem likely you are making a Type II error here?

Based on **"Fisher's significance testing" paradigm**:
4) Does there seem to be a difference between these groups?

5) Which paradigm do you think best gets at what is truly happening?

Answers:

1) Since the p-value is 0.055 and greater than α, the results are not statistically significant according to the Neyman and Pearson paradigm, and we fail to reject the null hypothesis.
2) No, it does not seem likely, since we did not reject the null hypothesis.
3) It seems likely that we would be making a Type II error because we fail to reject the null, but the p-value is still low and close to the significance level.
4) Based on Fisher's significance testing paradigm, since the p-value is low (0.055), there is evidence against the null hypothesis and it seems that there is a difference between the two groups.
5) It seems that Fisher's paradigm gets best at what is happening. It gives a more accurate conclusion given existing scientific literature, and the data we have does give strong evidence against the null despite the p-value being more than α.

Part 1.6 (5 points) As we also discussed in class (and the previous homework) R has several built in functions to do parametric hypothesis tests. In particular, R has a built in function called `t.test(sample1, sample2)` that takes two samples of data and runs a t-test on them. Use this function to compare the sleep and caffeine groups and report the p-value (which should be slightly different from the one you got when you did your own t-test above). Also, describe whether your conclusions would be different using this built in function compared to when you ran your t-test above if you a) followed the Neyman-Pearson's hypothesis testing paradigm, and b) if you followed Fisher's significance testing paradigm. Finally, describe the reason why your p-value differs from the p-value returned by the t.test() function.

```
t.test(sleep_condition, caffeine_condition)
```

```
##
##  Welch Two Sample t-test
##
## data:  sleep_condition and caffeine_condition
## t = 2.1438, df = 21.894, p-value = 0.04342
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.09699633 5.90300367
## sample estimates:
## mean of x mean of y
##     15.25     12.25
```

Answers:

The p-value using the t.test() function is 0.04342, which is less than the one calculated above. Now, according to the Neyman-Pearson paradigm, the results are statistically significant since the p-value is less than the alpha value, and we reject the null hypothesis (a different result from above). If we follow Fisher's paradigm there is also significant evidence against the null hypothesis as well since the p-value is small (the same conclusion as above).

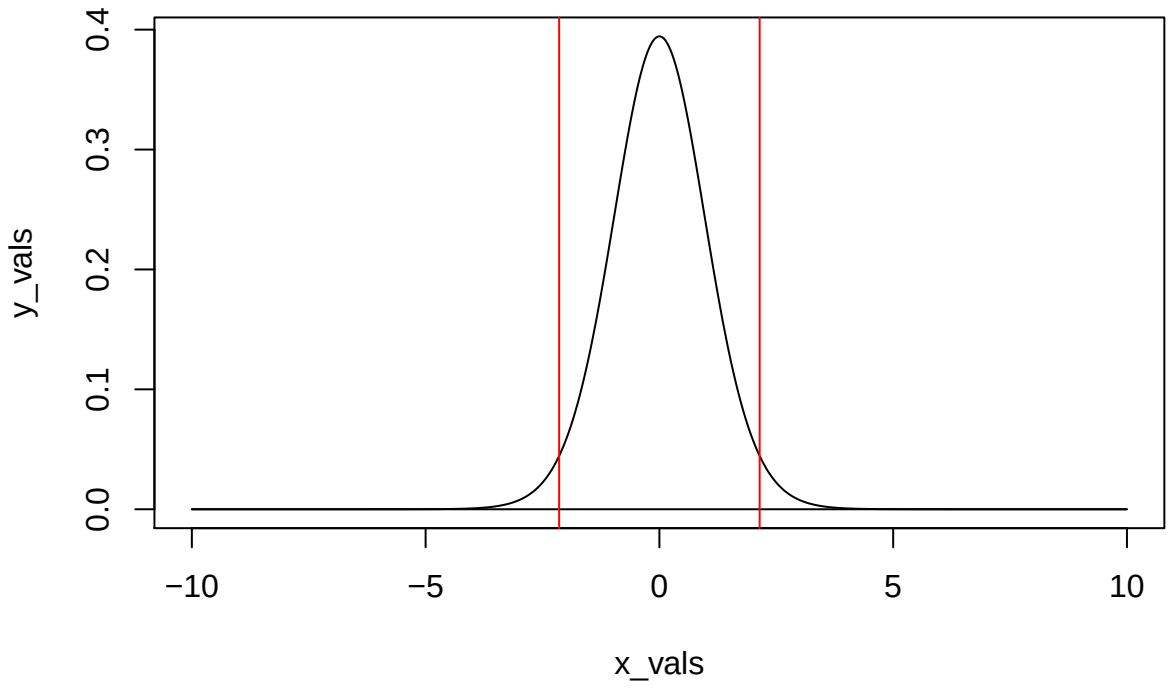
The difference in the two tests is due to a change in the degrees of freedom. Since df is greater in this test, the null distribution is more normally distributed and there is less area under the null distribution curve at the tail ends, meaning the same t-value will result in a smaller p-value when the degrees of freedom is higher.

Part 1.7 (7 points): Based on your answer in part 2.6, modify the t-test code you wrote to produce results that are consistent with R's t.test function.

```
# sample sizes for the two conditions
n1 <- length(sleep_condition)
n2 <- length(caffeine_condition)

# estimate the degrees of freedom
(df <- 21.894)
```

```
## [1] 21.894
```

```
# plot the null distribution with the observed statistic on it
x_vals <- seq(-10, 10, length.out = 1000)
y_vals <- dt(x_vals, df)

plot(x_vals, y_vals, type = "l")
points(x_vals, rep(0, length(x_vals)), type = 'l')
abline(v = obs_tstat, col = "red")
abline(v = -obs_tstat, col = "red")
```



```r
(p_val <- pt(obs_tstat, df = df, lower.tail = FALSE) + pt(-obs_tstat, df = df))
```

```
## [1] 0.04341568
```

Part 2.1 (12 points)

In class we discussed that the mathematical derivation of different parametric tests are based on a set of assumptions/conditions (For actual derivations for population tests see this link)[https://pdfs.semanticscholar.org/6297/58de27161160c5ce051a6736c8b2004b42bc.pdf].

In practice, the inferences for a t-test are usually valid when a set of "rules of thumb" have been met. For a t-test, these rules of thumb are that either: the data looks relatively normal (i.e., doesn't have any large outliers), or alternatively, the sample size is n $>$ 30.

Let's do a simulation to see how robust the t-test is. Start by creating a null vector called `p_values` and then write a for loop that runs 10,000 simulations, where in each simulation you:

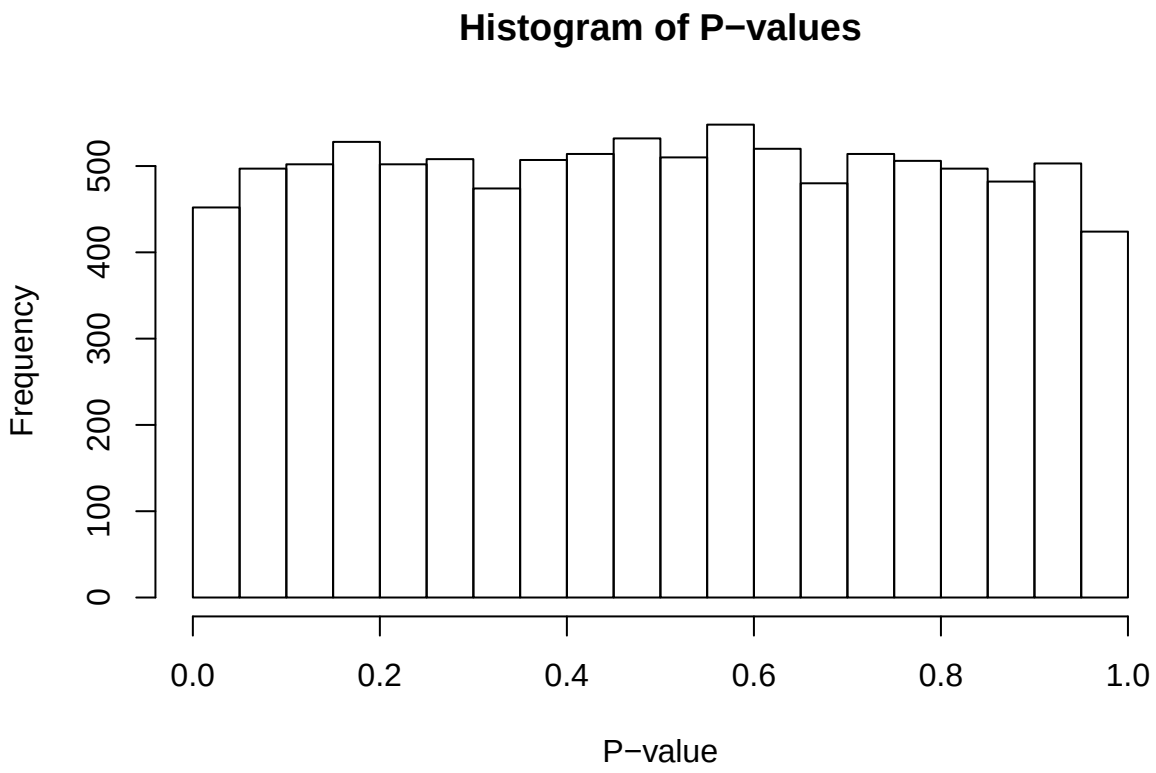
1) Draw a random sample of size n = 30 from an exponential distribution using the `get_sample(n)` function I have written. Save this sample in an object called `sample1`
2) Draw a second random sample also of size n = 30 using the `get_sample(n)` function. Save this sample in an object called `sample2`
3) Run a t-test and save the p-value for the t-test in the vector `p_values`

Once the for loop is done running, calculate the proportion of p-values that were less than $\alpha = 0.05$. Report whether it is the case that 5% or less of these p-values are indeed less than 0.05. Also plot a histogram of the p-values you collected and report the shape of this histogram.

```r
set.seed(123)
# a function that returns a sample of data of size n (from an exponential distribution)
get_sample <- function(n){
  rexp(n)
}

# write a for loop that 10,000 time, gets two samples of size n
# and calculate the p-value
p_values <- NULL
for (i in 1:10000){
  sample1 <- get_sample(30)
  sample2 <- get_sample(30)
  p_values[i] <- t.test(sample1, sample2)$p.value
}


# see if the percentage of significant p-values is what is expected based on the alpha level
mean(p_values <= 0.05)
```

```
## [1] 0.0452
```

```r
#histogram
hist(p_values, main = "Histogram of P-values",
     xlab = "P-value",
     ylab = "Frequency")
```



Answers:

The proportion of p-values less than 0.05 is 0.0452, which is indeed less than 5 percent. The histogram of p-values has a uniform distribution.

Bonus part 2.2 (0 points)

Try changing properties of the simulation above to see if you can 'break' the t-test, i.e., if you can get more than 5% of p-values to be less than the signifiance level of $\alpha = 0.05$. In particular, try changing the sample size n, and the underlying distribution of the data (i.e., change the `get_sample()` function). For the changes you make, always keep n to be at least 10, and do not use any if statements in your `get_sample()` function.

Part 3: Simple linear regression

In 2000, the United States presidential election was between a Yale alumnus, George W. Bush who was the Republican candidate, and a Harvard alumnus Al Gore who was the Democratic candidate. There were also

a number of "third-party" candidates such as Princeton alumnus Ralph Nader who was the Green Party candidate, and Georgetown alumnus Pat Buchanan who was the Reform Party candidate.

The code chunk below contains data from the 2000 election for the state of Florida in a data frame called `florida data`. Each observational unit in this data frame contains information from the 67 counties in Florida including demographic information on each county as well as the votes received by each candidate in each county.

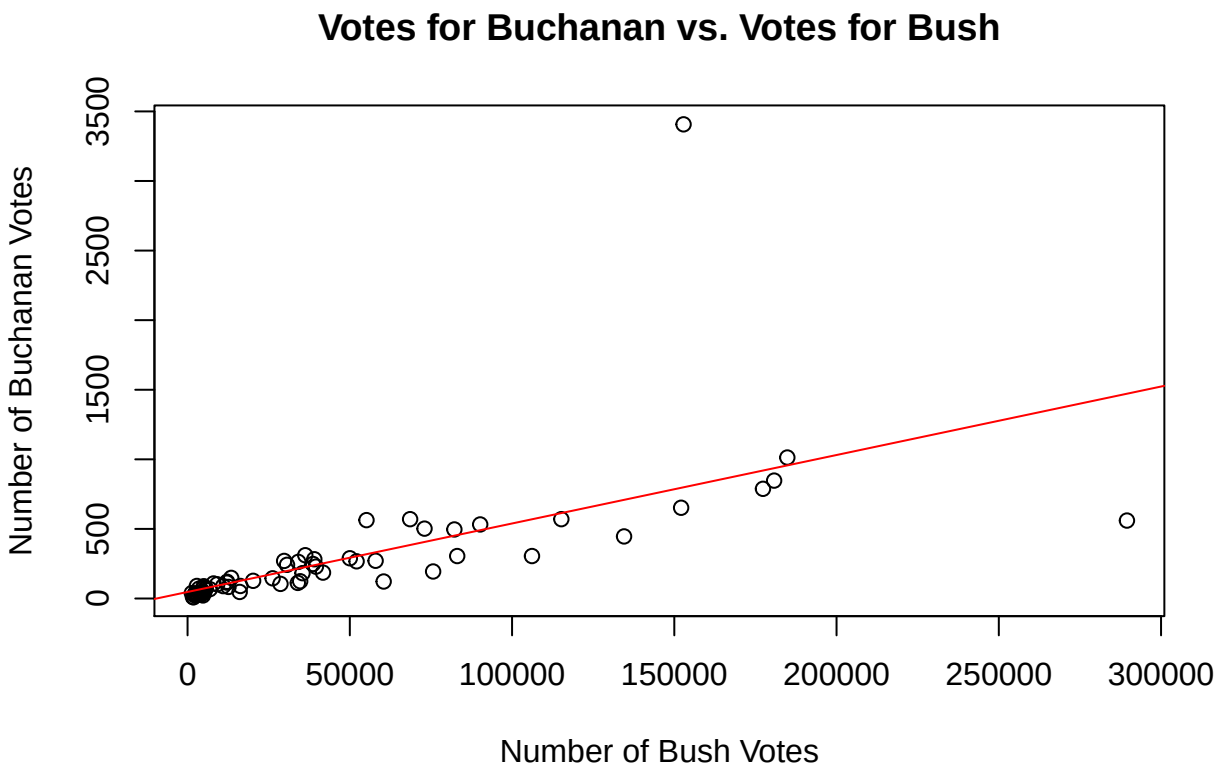
This the exercises below, you will use linear regression to look at the relationship between the votes that the Republican candidate *George W. Bush* received and the votes that the Reform candidate *Patrick Buchanan* received.

```
load('florida_vote_data_2000.Rda')
```

Part 3.1 (7 points):

Start the analysis by creating a scatter plot of the number of votes that Pat Buchanan recieved as a function of the votes that George Bush recieved. Then fit a linear model that can predict the number of votes Buchanan should receive given the number of votes that Bush received, and add the regression line to this plot in red.

```
plot(florida_data$Buchanan ~ florida_data$Bush,
     main = "Votes for Buchanan vs. Votes for Bush",
     xlab = "Number of Bush Votes",
     ylab = "Number of Buchanan Votes")
lm_fit <- lm(Buchanan ~ Bush, data = florida_data)
abline(lm_fit , col = "red")
```



Part 3.2 (7 points):

Now extract the coefficients from the linear model. In the space below report how many votes Buchanan is expected to get for every 1,000 votes Bush received, and how many votes the model predicts that Buchanan would have gotten if Bush had received 0 votes. Finally, write an equation that predicts the number of votes Buchanan should get as a function of the number of votes Bush received (make sure to use use $LaTeX$ for the proper notation).

```
coef(lm_fit)
```

```
##  (Intercept)         Bush
## 46.972816323  0.004920082
```

```
#Buchanan votes for every 1000 Bush votes
lm_fit$coefficients[2]*1000
```

```
##     Bush
## 4.920082
```

Answers:

To find the number of votes Buchanan is expected to get, we multiply the slope by 1000. So for every 1000 votes Bush gets, Buchanan is expected to get 4.92. If Bush had received 0 votes, Buchanan would have received $y = 46.97$ according to the model (the y-intercept).

Equation:

$\hat{y}$ = Predicted Number of votes Buchanan receives

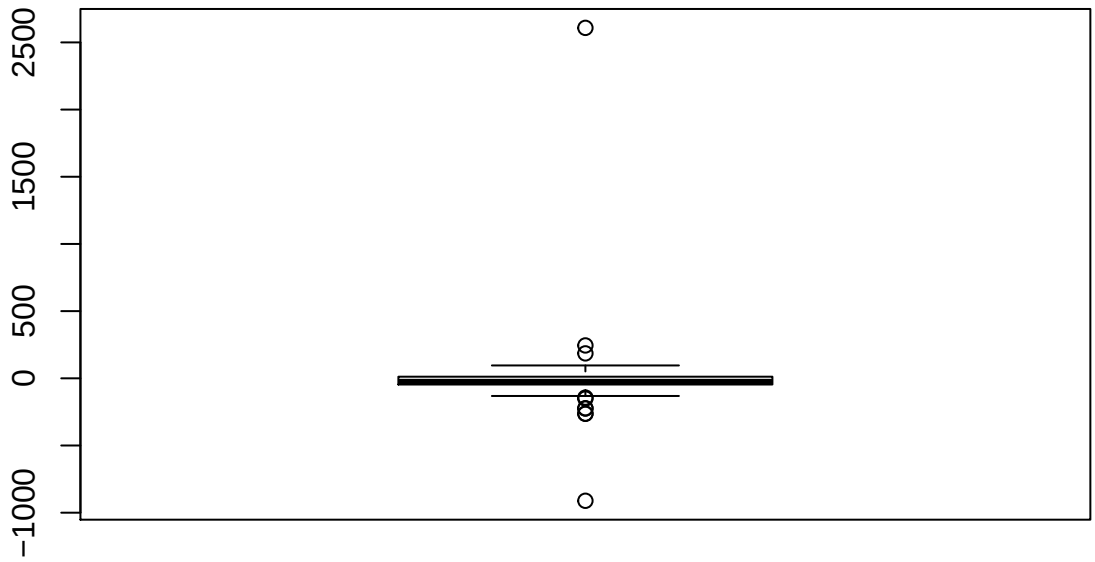
x = Number of votes Bush receives

$$\hat{y} = 0.00492x + 46.97$$

Part 3.3 (15 points):

From looking at the plot above, it should be clear that there is one extreme outlier. To see this more clearly, create a boxplot of the residuals of the model below. Then report: 1) what is the county that the outlier corresponds to, 2) how many votes Buchanan actually received in that county, 3) the predicted number of votes that Buchanan should have received for this county based on the regression model fit above for that county, and 4) the value of the residual for this county. Be sure to use the appropriate notation when reporting these numbers. Finally, use the Internet to come up with a reasonable explanation that could have led to this outlier (embedding images in the markdown document could be useful here).

```
# boxplot of the residuals
boxplot(lm_fit$residuals)
```




```
# county that is the outlier
#takes the county of the max residual
which.max(lm_fit$residuals)
```

```
## 50
## 50
```

```
florida_data[which.max(lm_fit$residuals),2]
```

```
## [1]   Palm Beach
## 67 Levels:   Alachua              ...   Washington
```

```
# actual number of Buchanan votes
florida_data$Buchanan[50]
```

```
## [1] 3407
```

```
# predicted number of Buchanan votes
lm_fit$fitted.values[50]
```

```
##       50
## 798.9876
```

```
# residual value
lm_fit$residuals[50]
```

```
##       50
## 2608.012
```

Answers

1) The outlier corresponds to Palm Beach county.
2) Buchanan received $y = 3407$ votes in Palm Beach.
3) The predicted number is $\hat{y} = 798.9876$.
4) The residual is $y - \hat{y} = 2608.012$.

According to the American Political Science Review, one possible explanation for this outlier is that the Palm Beach ballot, called a "butterfly ballot", was misleadingly designed such that Democrats mistakenly voted for Buchanan, thinking they were voting for Al Gore.

See Figure (on the next page) for a picture of the ballot.

Part 3.4 (7 points):

Suppose that Buchanan received exactly the number of votes predicted by the regression model, and the residual number of votes he received were intended to be votes for Al Gore. To examine the consequences of this, start by calculating the total number of votes Bush received and the total number of votes Gore received. Then add the residual number of Buchanan votes from the outlier county to the total number of votes that Gore received. Create an R Markdown table below and report these numbers. Would have this changed who got the majority number of votes (and hence who would have won Florida)?

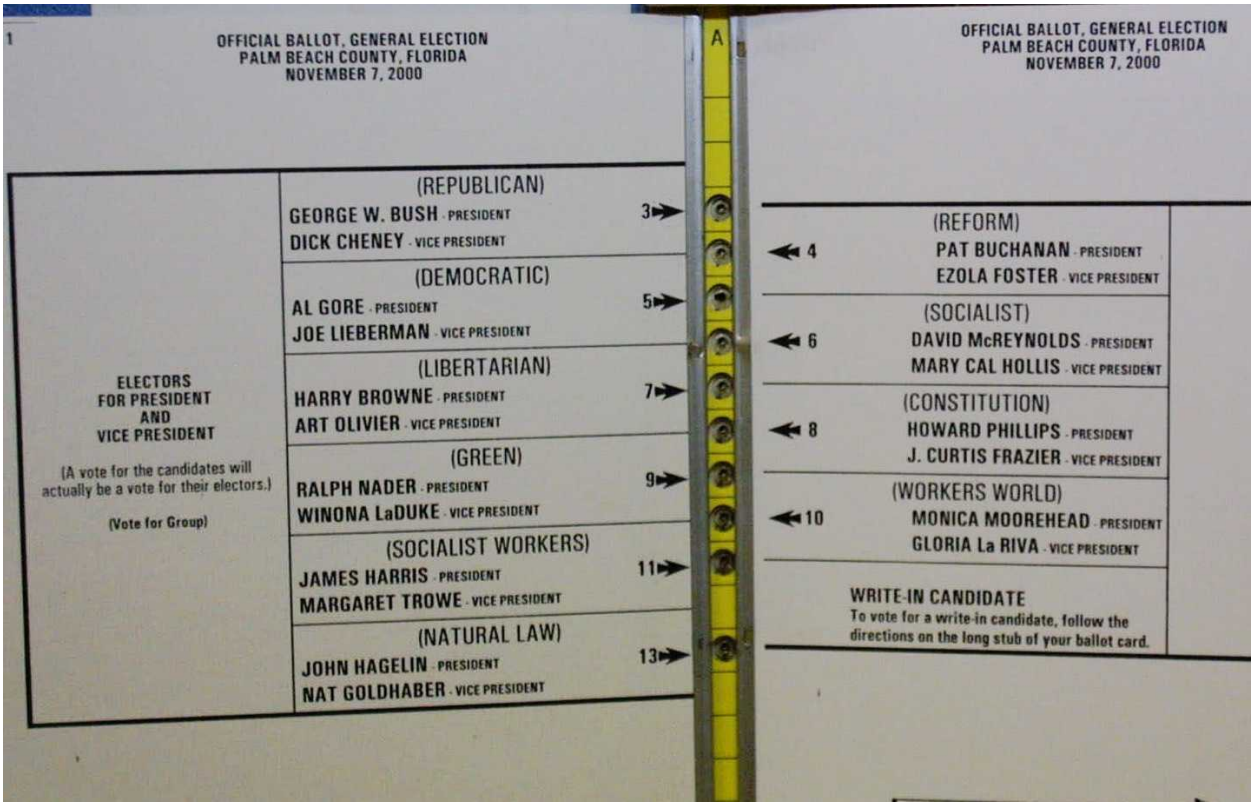


Figure 1: Picture of Butterfly Ballot

```
(bush <- sum(florida_data$Bush))
```

```
## [1] 2910078
```

```
(gore <- sum(florida_data$Gore))
```

```
## [1] 2909117
```

```
(gore_out <- sum(florida_data$Gore) + lm_fit$residuals[50])
```

```
##      50
## 2911725
```

Answers

Bush Total	Gore	Gore Total with Outlier
2910078	2909117	2911725.012387

Based on these numbers, if the residual votes were added to Al Gore's total, then Gore would have won Florida, not Bush.

Part 3.5 (2 points):

The United States uses the Electoral College system. In this system, the candidate who got the majority of the vote in a state wins all the Electoral College votes for that state (at least for most of the states in the US including Florida). Use the Internet to find the number of votes that Bush won the Electoral College

Homework 7

The purpose of this homework is to practice wrangling data and conducting inference for simple linear regression models. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday November 3rd.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Part 1: Data wrangling with dplyr

On July 3rd 2015, my 1999 Toyota Corolla broke down on the side of the highway outside of Sturbridge MA. While I had the car repaired, I knew it was time to sell it and get a new car. I intended to sell my Corolla to the car dealership, the only catch was that I was not sure how much the used Corolla was worth. In the following excercises we will model how much a used Corolla is worth as a function of the number of miles it has been driven.

The data we will look at comes from Edmunds.com which is a website where you can buy new and used cars online. This data set is from the 2015 DataFest competition, which is an undegraduate data science competition that takes place at difference colleges across the United States. The data has been made available to this class for educational purposes, however please do not share this data outside of the class.

Part 1.1 (15 points): Let's start by loading the dplyr libary and data set using the code below. Report how many cases and variables the full data set has. Then use the dplyr select() and filter() functions to create a reduced data frame object called `used_corollas` in which:

1) The only variables in that should be in the `used_corollas` data frame are:

a) model_bought: the model of the car
b) new_or_used_bought: whether a car was new or used when it was purchased
c) price_bought: the price the car was purchased for
d) mileage_bought: the number of miles the car had when it was purchased

2) The only cases should be in the `used_corollas` data frame are:

a) used cars
b) Toyota Corollas
c) cars that have been drive less than 150,000 miles

3) Finally use the na.omit() function on the `used_corollas` data frame to remove cases that have missing values.

If you have properly filter the data, the resulting data set should have 248 cases, so check this is the case before going on to the next set of exercises.

```
# load the dplyr library
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# load the data set
load("car_transactions.rda")


# get the size of the original data set
dim(car_transactions)
```

```
## [1] 107832     21
```

```
#glimpse(car_transactions)

#use dplyr to reduce the data set to only used Corolla's with under 150,000 miles
used_corollas <- filter(car_transactions, model_bought == "Corolla",
                        new_or_used_bought == "U", mileage_bought < 150000)
used_corollas <- select(used_corollas, model_bought,
                        new_or_used_bought, mileage_bought, price_bought)
used_corollas <- na.omit(used_corollas)

# check the size of the resulting data frame
dim(used_corollas)
```

```
## [1] 248   4
```

Answers

There are 21 variables and 107832 observations in the original data set. There are 248 cases and 4 variables in the filtered data set.

Part 2: Fitting a linear model and statistical inference on regression coefficients

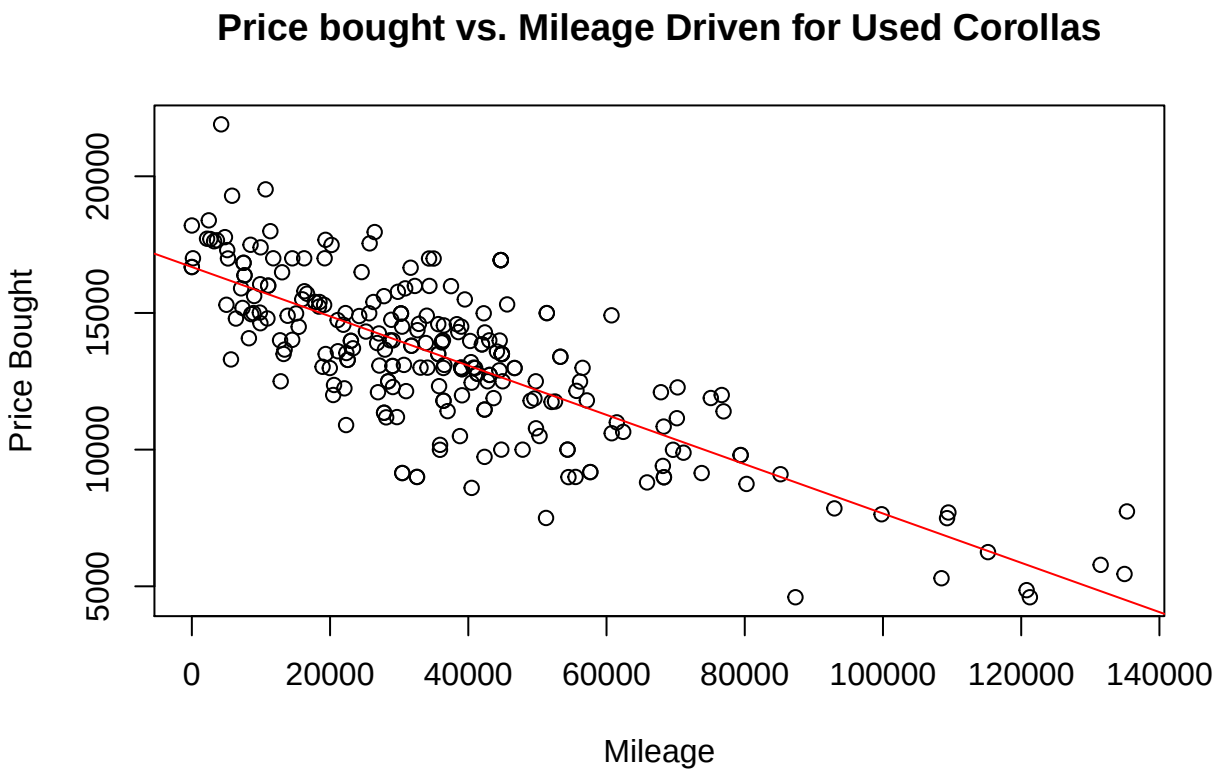
Now that we have the relevant data, let's examine the relationship between a car's price and the number of miles driven!

Part 2.1 (10 points): Let's begin analyzing the data by taking the following steps:

1) Plot the price as a function of the number of miles driven.
2) Fit a linear model regression model that shows the predicted (expected) price as a function of the number of miles driven. Save this model to an object called `lm_fit` which you will use throughout the rest of this homework.
3) Add a red line to our plot showing the regression line fit.
4) Print the regression coefficients found.

Report how much does the price of a Corolla decrease for every additional mile it has been driven, and also what this regression model suggests a car that has been driven 0 miles would be worth. Finally, write out the regression equation.

```
# let's start by plotting the data
plot(used_corollas$mileage_bought, used_corollas$price_bought,
     main = "Price bought vs. Mileage Driven for Used Corollas",
     xlab = "Mileage", ylab = "Price Bought")


# fit a regression model  (note: this is y as as function of x)
lm_fit <- lm(price_bought ~ mileage_bought, data = used_corollas)

# add the regression line to the plot
abline(lm_fit, col = "red")
```




```
# print the regression coefficients
coef(lm_fit)
```

```
##    (Intercept) mileage_bought
## 16681.91992781    -0.09018627
```

Answers:

The price of a Corolla decreases −$0.09 for every additional mile driven. The model suggests a used car at 0 miles would be worth $16681.92.

The regression equation is:

x = miles driven

$\hat{y}$ = predicted price of Corolla

$\hat{y} = -0.0902x + 16681.92$

Part 2.2 (5 points): Now use R's summary() function to report whether there is statistically significant evidence that the price of a car decreases as a function of the number of miles driven. Also, write out the hypothesis that is being tested using the appropriate symbols/notation discussed in class.

```
# get information about the statistical significance of the fit
summary(lm_fit)
```

```
##
## Call:
## lm(formula = price_bought ~ mileage_bought, data = used_corollas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4791.8 -1131.9    -0.3  1027.7  5600.7
##
## Coefficients:
##                    Estimate   Std. Error t value            Pr(>|t|)
## (Intercept)    16681.919928   204.459353   81.59 <0.0000000000000002 ***
## mileage_bought    -0.090186     0.004539  -19.87 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1816 on 246 degrees of freedom
## Multiple R-squared:  0.616,  Adjusted R-squared:  0.6145
## F-statistic: 394.7 on 1 and 246 DF,  p-value: < 0.00000000000000022
```

```
(p_value <- summary(lm_fit)$coefficients[2,4])
```

```
## [1] 0.00000000000000000000000000000000000000000000000000004747174
```

Answer

$H_0 : \beta_1 = 0$

$H_A : \beta_1 \neq 0$

$\alpha = 0.05$

The p-value is very close to 0 and less than our signficance level, so we reject the null hypothesis. There is statistically significant evidence of correlation between mileage and price.

Part 2.3 (5 points): We can create confidence intervals using a t-distribution via the confint() function. Report what the confidence interval for slope of the regression line is. Also, based on the confidence interval, explain why it it seems likely that the price of a car is not independent of the number of miles driven.

```
confint(lm_fit)
```

```
##                        2.5 %         97.5 %
## (Intercept)    16279.20570876 17084.63414685
## mileage_bought    -0.09912747    -0.08124508
```

Answer

The confidence interval is [-0.09912747, -0.08124508]. Since this interval does not include 0 which would imply no correlation (and we know it has a 95% probability of containing the true parameter), it seems likely that the price of a car is not independent of the number of miles driven.

Part 2.4 (10 points): We can also use the bootstrap to create confidence intervals for the slope of the regression coefficient. To do this you can use the following procedure:

1) Create a bootstrap resampled data frame by sampling with replacement from the `used_corollas` data frame. You can do this using dplyr's sample_n() function with the sample size being the number of cases in the `used_corollas` data frame and setting the `replace = TRUE` argument.

2) Fit the regression model using the bootstrap data frame.

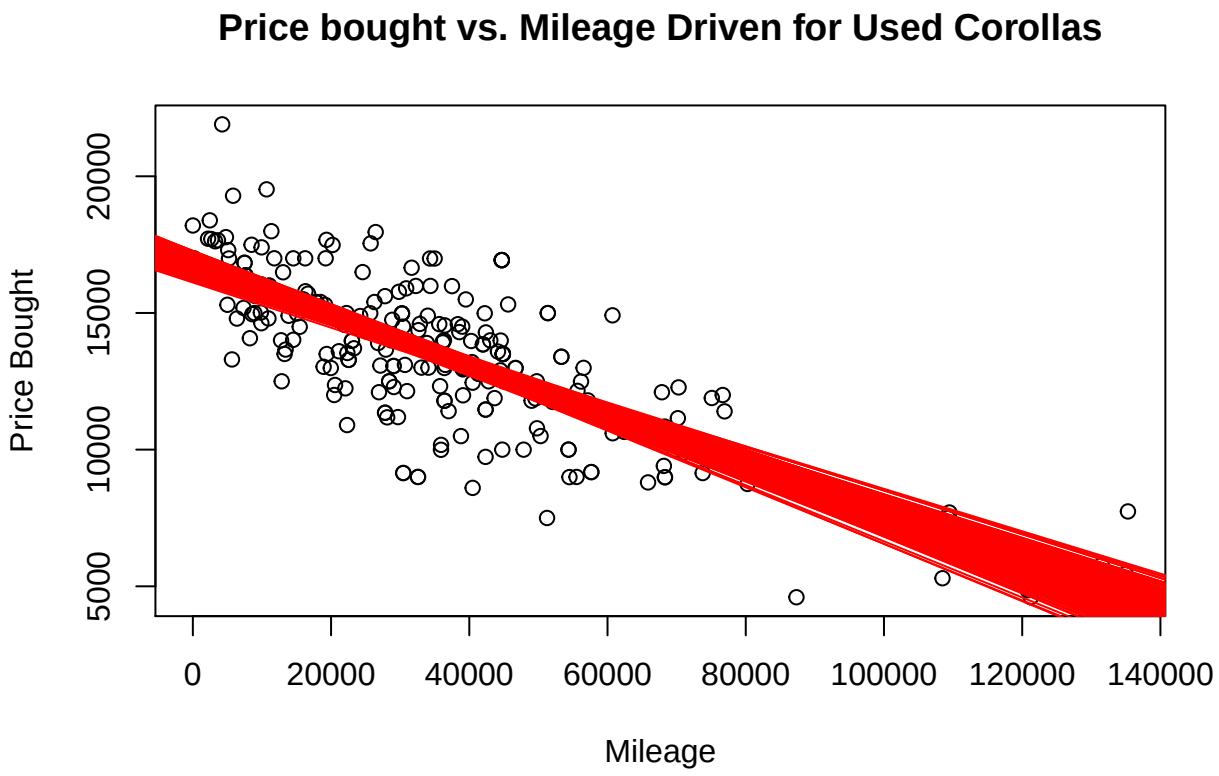
3) Extract the regression slope coefficient and save it to a vector object.

4) Repeat this process 1,000 times (this is less than normal because it is computationally expensive).

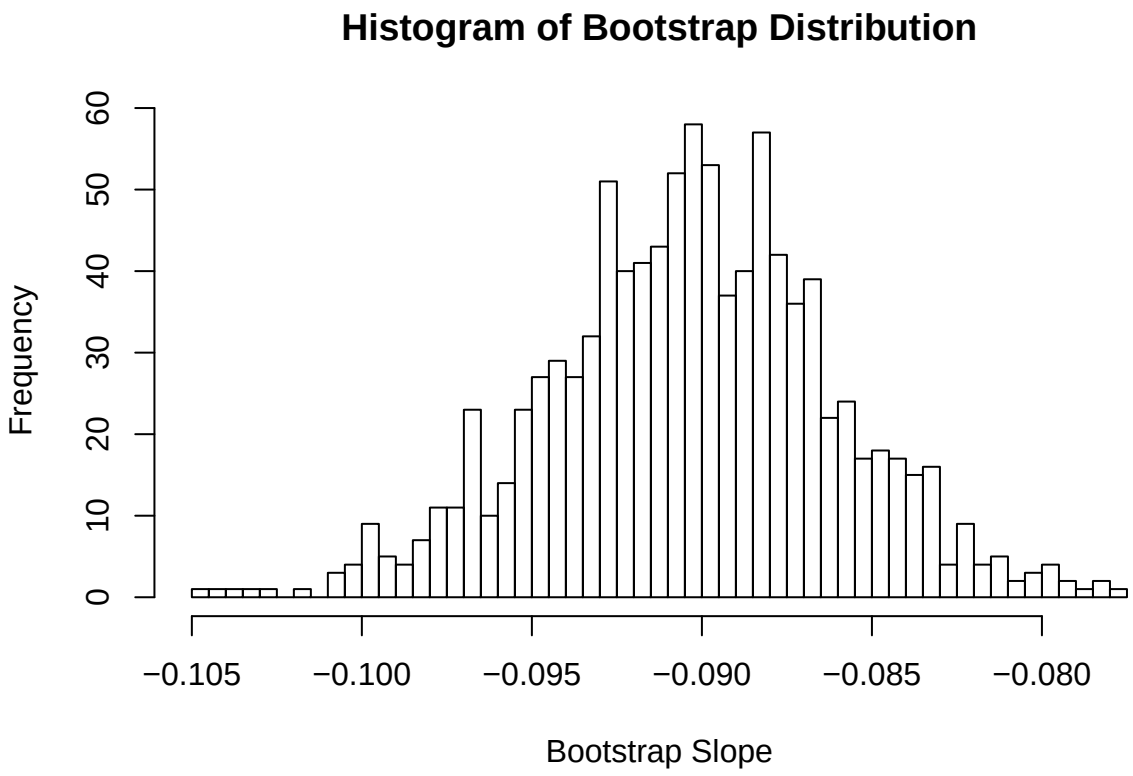
5) Use the percentile method to report a 95% confidence interval for the regression slope.

Also report whether the bootstrap confidence interval is similar to the confidence interval using the t-distribution you calculated above.

```
plot(used_corollas$mileage_bought, used_corollas$price_bought,
     main = "Price bought vs. Mileage Driven for Used Corollas",
     xlab = "Mileage", ylab = "Price Bought")

# create the bootstrap distribution
nrep <- 1000
n_cases <- dim(used_corollas)[1]
result_vec <- rep(0, nrep)
for (i in 1:nrep){
  boot_sample <- dplyr::sample_n(used_corollas, size = n_cases, replace = TRUE)
  boot_fit <- lm(price_bought ~ mileage_bought, data = boot_sample)
  abline(boot_fit, col = "red")
  result_vec[i] <- coef(boot_fit)[2]
}
```




```r
# plot it
hist(result_vec, main = "Histogram of Bootstrap Distribution",
     xlab = "Bootstrap Slope", ylab = "Frequency", nclass = 50)
```




```
# create bootstrap confidence intervals
boot_conf <- quantile(result_vec, c(0.025, 0.975))
boot_conf
```

```
##        2.5%       97.5%
## -0.09905251 -0.08216317
```

Answers:

The 95% Confidence interval is [-0.0990525, -0.0821632]. It is similar to the one calculated using the t-distribution, and both do not include 0.

Part 2.5 (10 points): My Toyota has 180,000 miles at the time I wanted to sell it. Based on the regression model fit above, what is the predicted worth of this car? Does this seem like a reasonable estimate?

```
#Worth of car calculated using predict() function
predict(lm_fit, newdata = data.frame(mileage_bought = 180000))
```

```
##        1
## 448.3911
```

```
#Worth of the car calculated using the above regression equation
16681.91993 - 0.09019*180000
```

```
## [1] 447.7199
```

Answer The predicted worth of the car using the predict() function is $448.39. I would say this estimate is a bit too low since even if the mileage is high a car would rarely be sold for so little. It might be possible to sell at this price since the car has already broken down on the highway before, but based on other prices online, a similar car should go for aroun $2000 or more. The low estimate is probably due to extrapolation from the line since the mileage we are given in our dataset does not exceed 140000.

Part 3: Analysis of variance (ANOVA) for regression

As discussed in class, we can also use an ANOVA to test the regression coefficients. We will explore relationship between the ANOVA and other analysis methods below.

Part 3.1 (5 points): Let's look at the relationship between the ANOVA F-statistic and the t-statistic. Use the anova() function on the linear model you fit and print out the ANOVA table. Look back to question 2.2 and create an object called `t_stat` that has the t-value that you was obtained from using the summary() function to get the t-statistic for the regression slope. Show that this the value of `t_stat` squared is (approximately) equal to the F-statistic found with the anova() function by printing both the t^2 value and the F-statistic value. Also, report the value of the sum of the model sum of squares (SSModel) and the residual sum of squares (SSError).

```
anova(lm_fit)
```

```
## Analysis of Variance Table
##
## Response: price_bought
##                  Df     Sum Sq    Mean Sq F value                Pr(>F)
## mileage_bought    1 1302085889 1302085889   394.7 < 0.00000000000000022 ***
## Residuals       246  811530941    3298906
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#T-value squared from summary function
t_stat <- -19.86712
(t_stat <- t_stat^2)
```

```
## [1] 394.7025
```

```
#F-statistic
anova(lm_fit)$"F value"[1]
```

```
## [1] 394.7023
```

```r
#SSModel
(ssmodel <- anova(lm_fit)$"Sum Sq"[1])
```

```
## [1] 1302085889
```

```r
#SSError
(sserror <- anova(lm_fit)$"Sum Sq"[2])
```

```
## [1] 811530941
```

```r
(sstotal <- ssmodel + sserror)
```

```
## [1] 2113616830
```

Answers:
As shown above, the t-value squared and F-statistic are approximately equal (394.7025 and 394.7023 respectively). SSModel is 1302085889.42664 and SSError is 811530940.763768. SSTotal is 2113616830.1904.

Part 3.2 (10 points): We can also extract the SSModel, SSError and SSTotal using values from the original data and from values stored in the `lm_fit` object. Run the following analyses to to calculate the SSModel, SSError and SSTotal values:

1) For the SSTotal, use the `used_corolla` data frame to calculate $\Sigma_{i=1}^{n}(y_i - \bar{y})^2$.
2) Use the `fitted.values` in the `lm_fit` object to caluclate SSModel using the formula $\Sigma_{i=1}^{n}(\hat{y}_i - \bar{y})^2$.
3) Use the `residuals` in the `lm_fit` object to caluclate SSError using the formula $\Sigma_{i=1}^{n}(\hat{y}_i - y_i)^2$.

To check you have the right answers, look at the values you got in question 3.1. Show that SSTotal = SSModel + SSError for the values you calculated.

```r
# the total sum of squares
(SSTotal <- (sum((used_corollas$price_bought - mean(used_corollas$price_bought))^2)))
```

```
## [1] 2113616830
```

```r
# the model sum of squares
(SSModel <- (sum((lm_fit$fitted.values - mean(used_corollas$price_bought))^2)))
```

```
## [1] 1302085889
```

```r
# the sum of squared error
(SSError <- sum(lm_fit$residuals^2))
```

```
## [1] 811530941
```

```
# show that SSTotal is equal to SSModel + SSError
SSError + SSModel
```

```
## [1] 2113616830
```

```
SSTotal
```

```
## [1] 2113616830
```

As we can see SSTotal = SSModel + SSError.

Part 3.3 (5 points): We also discussed in class that for simple linear regression, correlation coefficient squared (r^2) is equal to the percentage of the variance explained by the liner model: SSModel/SSTotal. Calculate the correlation coefficient between mileage_bought and price_bought, and square it (which gives you the *coefficient of determination*). Then using the values calculated in part 3.2, show that this is equal to SSModel/SSTotal, and also equal to 1 - SSResidual/SSTotal.

```
(coefdet <- (cor(used_corollas$mileage_bought, used_corollas$price_bought))^2)
```

```
## [1] 0.6160463
```

```
SSModel/SSTotal
```

```
## [1] 0.6160463
```

```
1 - SSError/SSTotal
```

```
## [1] 0.6160463
```

As we can see these are all equal.

Part 4: Regression diagnostics

When making inferences about regression coefficients, there are a number of assumptions that need to be met to make these tests/confidence invervals valid. The assumptions for an ANOVA are:

1) **Normality**: residuals are normally distributed around the predicted value $\hat{y}$
2) **Homoscedasticity**: constant variance over the whole range of x values
3) **Linearity**: A line can describe the relationship between x and y
4) **Independence**: each data point is independent from the other points

To check whether these assumptions seem to be met by creating a set of diagnostic plots.

Part 4.1 (5 points): To check whether the residuals are normally distributed we can use create a Q-Q plot. The `car` package has a nice function to create these plots called qqPlot() to create these plots. If we pass the `lm_fit` object to the qqPlot() function it will create a Q-Q plot of the studentized residuals. Create this plot and report if the residuals seem normally distributed?

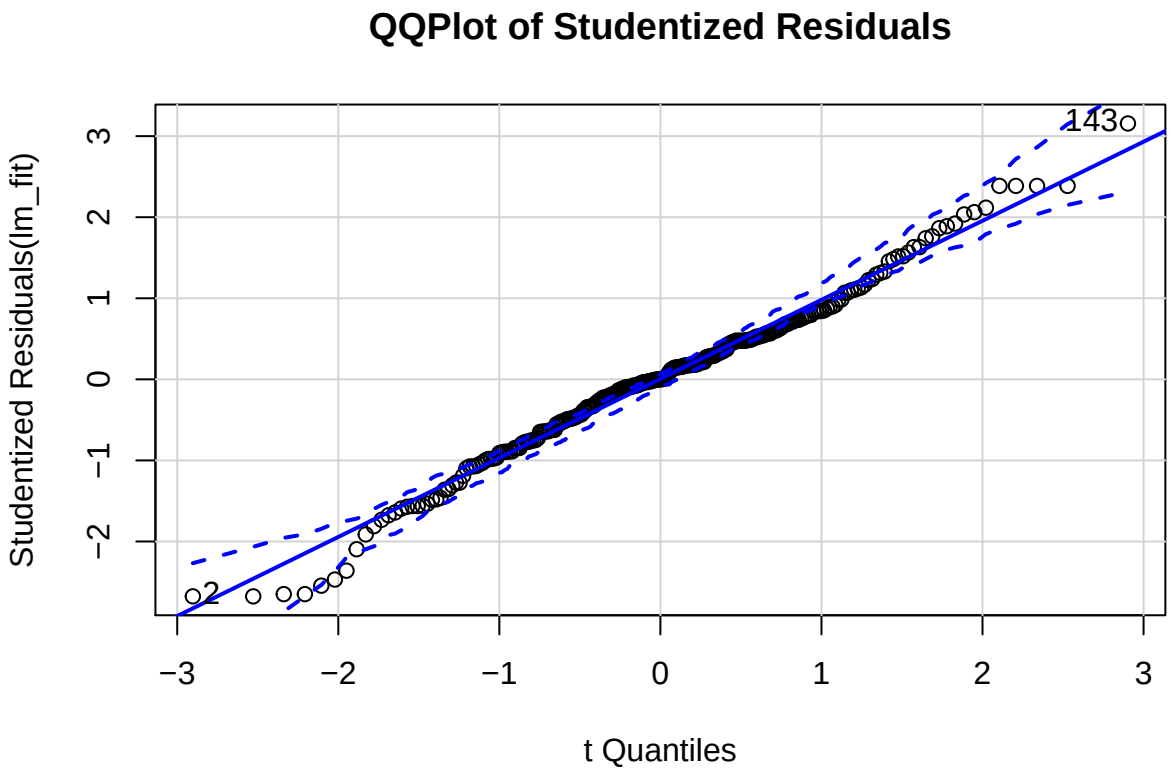
```
# install.packages('car')
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
qqPlot(lm_fit, main = "QQPlot of Studentized Residuals")
```



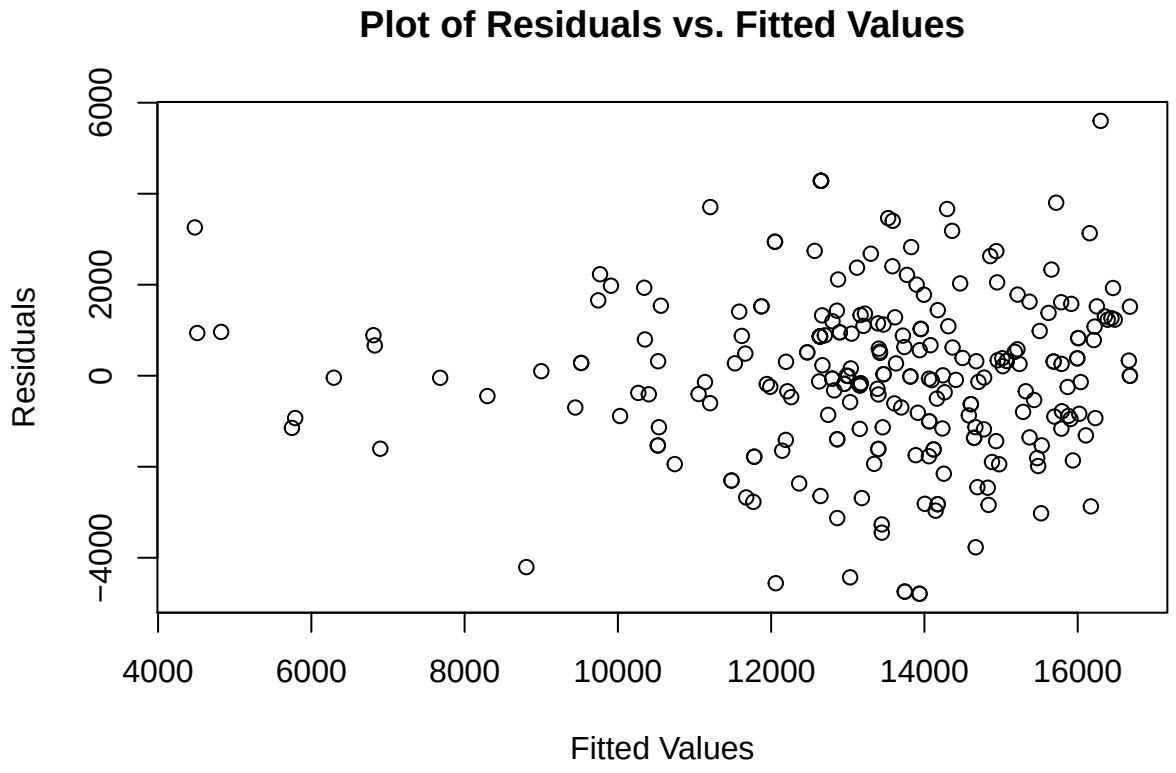

```
##   2 143
##   2 140
```

Answer:

The residuals are very close to the linear line y = x despite some variation at either end of the line, so they appear to be normally distributed.

Part 4.2 (5 points): To check for homoscedasticity and linearity, we can create a plot of the residuals as a function of the fitted values. Create such a plot below using information in the lm_fit object. Does it appear that homoscedasticity and linearity are met here? Are these results what you would expect from looking at plots above and from the nature of the type of data you are analyzing?

```
plot(lm_fit$fitted.values, lm_fit$residuals,
     main = "Plot of Residuals vs. Fitted Values",
     xlab = "Fitted Values", ylab = "Residuals")
```



Answers:

It is difficult to tell here, but it seems that the residuals seem to vary more as the fitted values increase, which shows that the plot exhibits some heteroscedasticity. This might be because cars with higher mileages all depreciate in a similar way, whereas cars with lower mileages may carry more varying values depending on other features of the car.

However, the residuals also seem to be centered around 0 with no particular pattern suggesting non-linearity. So based off of this plot I would say the linear model still stands. I originally thought that since the price of a used car should not go below 0, using a linear model to fit the data may not be the best choice - instead, a model like an inverse graph may be better if we want to predict the price of cars with very high mileage.

But based on the data frame we are given, where the mileage is still relatively low, a linear model still seems to work.

Part 4.3 (5 points): To check if the data points are independent requires knowledge of how the data was collected. For example, if the data you have is from a time-series (e.g., recordings of the temperature in New Haven on consecutive days) then there is a high likelihood that the data points might not be independent. On the other hand, if you take a simple random sample from a population where every point is equally likely to be selected, then the data is going to be independent.

Unfortunately I do not know exactly how this data was collected so it is difficult to say if the data is independent here. However, there might be ways to investigate whether it seems plausbile that it could be independent. Please describe some ways you might investigate whether the data could be independent (hint: think about the varaibles in the full `car_transactions` data set) Note: there is no exact 'right answer' here, just describe some possible ideas.

Answer:

To investigate whether or not the data is independent we could look at the dates and locations the cars were sold (date_sold and dma_bought, or another indicator of date and location). It is possible that some particular dealerships might be offering higher or lower prices on average than others, and if many cars in the dataset we are given were all sold at the same dealership, this may be reason to question the independence of the data points. The same goes for the state in which the car was sold. Similarly, perhaps during a certain period in time the value of a particular make/model of a car was increased or decreased, which would affect the independence of the data. For example, cars usually sell for higher around the holidays but then drop in price after the new year.

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 7

Homework 8

The purpose of this homework is to practice examining unusual points in simple linear regression models and to work with multiple regression models. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday November 10th.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Part 1: Preparing the data

Let's continue to examine Toyota Corolla data using the data set from Edmunds.com in order to gain practice examininig unusual points. Remember, the data has been made available to this class for educational purposes, so please do not share this data outside of the class.

Part 1.1 (5 points): Let's start again by loading the dplyr libary and the Edmunds data set using the code below. Again use the dplyr select() and filter() functions to create a reduced data frame object called `used_corollas_all`. However this time **do not do any filtering related to the number of miles a car has been driven** (i.e., keep in the data set cars that have been drive more than 150,000 miles).

In particular, follow the steps below:

1) The only variables that should be in the `used_corollas_all` data frame are:

a) model_bought: the model of the car
b) new_or_used_bought: whether a car was new or used when it was purchased
c) price_bought: the price the car was purchased for

2) The only cases should be in the `used_corollas_all` data frame are:

a) used cars
b) Toyota Corollas

3) Use the na.omit() function on the `used_corollas_all` data frame to remove cases that have missing values.

If you have properly filter the data, the `used_corollas_all` should have 249 cases, so check this is the case before going on to the next set of exercises.

Finally, recreate the used Corolla data set which you created in homework 7 that only includes cars that have been driven less than 150,000 miles, and save it in an object called `used_corollas_150k`. You should be able to create this data frame using just one line of R code once you have created the `used_corollas_all` data frame (and as you saw on homework 7, this data frame should have 248 cases).

```r
# load the dplyr library
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# load the data set
load("car_transactions.rda")

# use dplyr to created used_corollas_all that only has used Corollas
used_corollas_all <- filter(car_transactions, new_or_used_bought == "U",
                            make_bought == "Toyota", model_bought == "Corolla")
used_corollas_all <- select(used_corollas_all, model_bought, new_or_used_bought,
                            price_bought, mileage_bought)
used_corollas_all <- na.omit(used_corollas_all)


# check the size of the resulting data frame
dim(used_corollas_all)
```

```
## [1] 249   4
```

```r
# created the used_corollas_150k that contains only cars with less than 150,000 miles
used_corollas_150k <- used_corollas_all %>% filter(mileage_bought <= 150000)
dim(used_corollas_150k)
```

```
## [1] 248   4
```

Part 1.2 (10 points):

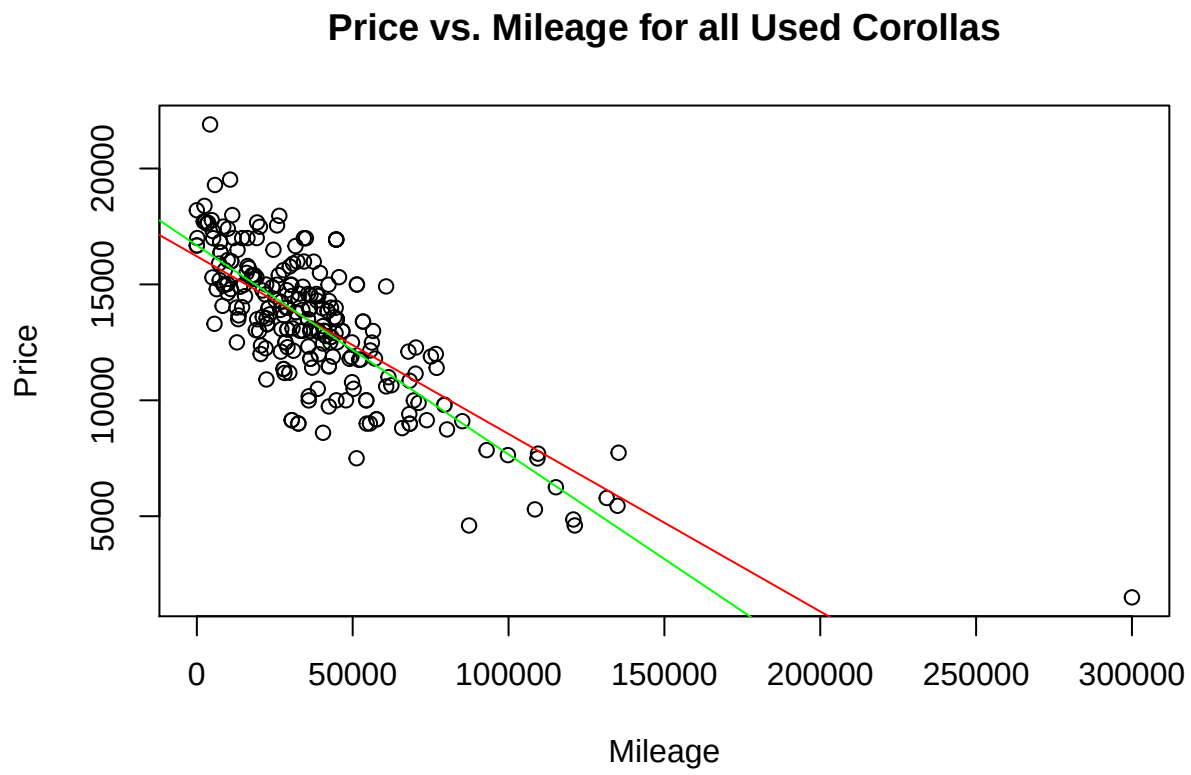
Now fit a linear regression model to the `used_corollas_all` that shows the predicted (expected) price as a function of the number of miles driven. Save this model to an object called `lm_fit_all`. Also, create a scatter plot of the price as a function of the number of miles driven, add a red line to our plot showing the regression line, and print the regression coefficients found.

Also, fit the regression model using the `used_corollas_150k` (as you did on homework 7). Add a green line to the plot showing the fit when the one additional data point is added, and describe below how similar the slope coefficient $\hat{\beta}_1$ is between these models. Finally, make a prediction for the price of a car that has been driven 150,000 miles using both the `lm_fit_all` and the `lm_fit_150k` models. Do these models seem similar to you?

```r
# start by plotting the data
plot(used_corollas_all$mileage_bought, used_corollas_all$price_bought,
     xlab = "Mileage", ylab = "Price", main = "Price vs. Mileage for all Used Corollas")


# fit a regression model, add the regression line to the plot and display the regression coefficients
lm_fit_all <- lm(price_bought ~ mileage_bought, data = used_corollas_all)
abline(lm_fit_all, col = "red")
coef(lm_fit_all)
```

```
##    (Intercept) mileage_bought
## 16210.25517005    -0.07660694
```

```r
# fit the model that is excluding the 150k data point
lm_fit_150k <- lm(price_bought ~ mileage_bought, data = used_corollas_150k)
abline(lm_fit_150k, col = "green")
```




```r
coef(lm_fit_150k)
```

```
##    (Intercept) mileage_bought
## 16681.91992781    -0.09018627
```

```r
# make a prediction for a car driven 150k miles using both models
predict(lm_fit_all, newdata = data.frame(mileage_bought = 150000))
```

```
##        1
## 4719.215
```

```r
predict(lm_fit_150k, newdata = data.frame(mileage_bought = 150000))
```

```
##        1
## 3153.979
```

Answers: $\hat{\beta}_1$ changes from -0.07660694 to -0.09018627 when the one point in excluded from the dataset. These two slope coefficients seem to be pretty similar but would result in large differences at higher mileages. At 150000 miles the data including the point predicts a price of $4719.215 while the data excluding the point predicts $3153.979. These results are pretty far apart so I would say the models are not very similar.

Part 1.3 (5 points): Create a 95% confidence interval for the value of the regression slope β_1 using the `used_corollas_150k`. If we were assuming that the confidence interval from the `used_corollas_150k` was reasonable, would the value for the regression slope found in the `used_corollas_all` model seem like a plausible value for what the true parameter value β_1 is (at the $\alpha = 0.05$ level)?

```r
(CI_150k <- confint(lm_fit_150k))
```

```
##                         2.5 %         97.5 %
## (Intercept)    16279.20570876 17084.63414685
## mileage_bought    -0.09912747    -0.08124508
```

Answer At $\alpha = 0.05$, the value for the regression slope found in `used_cororllas_all` would not be a plausible value for the true parameter since it is outside the confidence interval [-0.09912747, -0.08124508]. -0.0766 is not within this interval.

Part 1.4 (10 points): Now sort the data frame `used_corollas_all` so that the rows are in the order from smallest number of miles driven to the most number of miles driven, and store it again the same object called `used_corollas_all`. Refit the `lm_fit_all` using this sorted data frame (as a sanity check, the coefficients found should be the same as before). Then, recreate the scatter plot based on this sorted `used_corollas_all` data and add to this plot both the confidence intervals for **the regression line** in green and the prediction interval in blue (again using this sorted `used_corollas_all`). Describe how the equation for **the confidence interval for the regression line** leads to the fact that these confidence intervals becomes wider for cars that have been driven the most miles.

```r
# arrange the data and refit the model
used_corollas_all <- arrange(used_corollas_all, mileage_bought)
lm_fit_all <- lm(price_bought ~ mileage_bought, data = used_corollas_all)
coef(lm_fit_all) #We see these coefficients are the same as before
```

```
##    (Intercept) mileage_bought
## 16210.25517005    -0.07660694
```

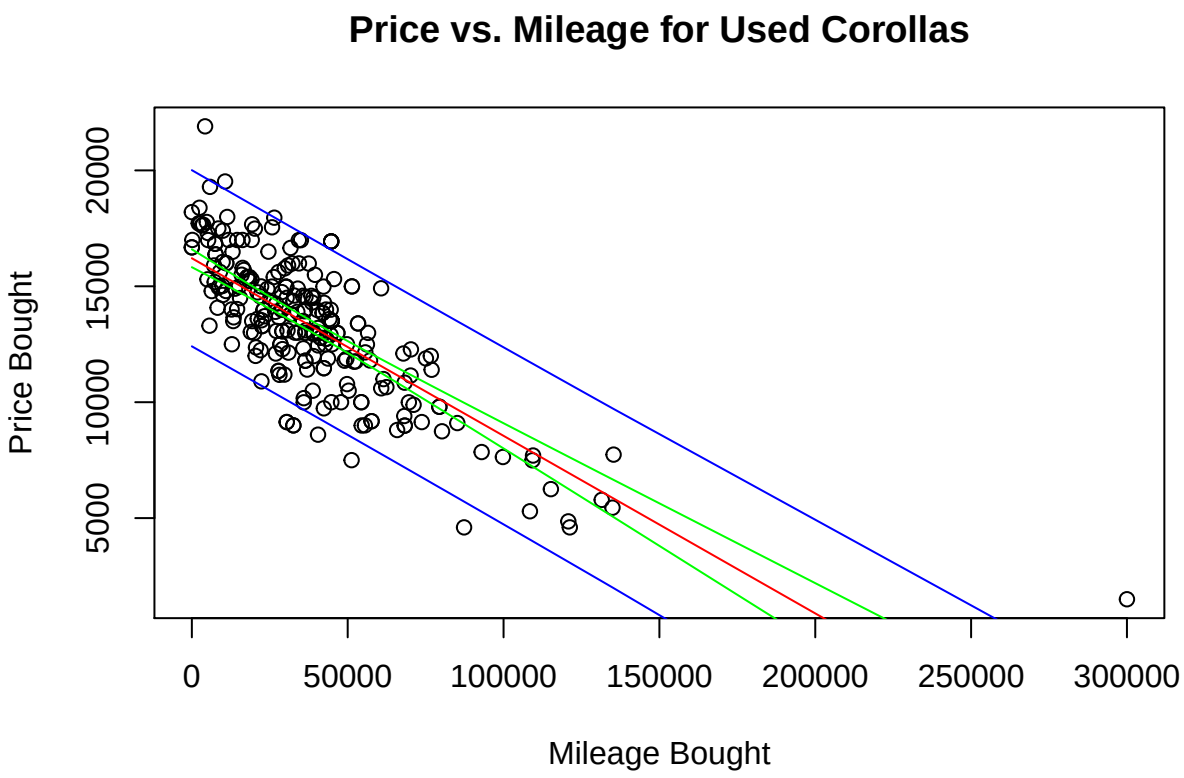
```r
# create confidence intervals for the betas
(CI_betas <- confint(lm_fit_all))
```

```
##                         2.5 %         97.5 %
## (Intercept)    15824.63955905 16595.87078104
## mileage_bought    -0.08450809    -0.06870578
```

```r
# create confidence interval for the regression line mu_y
CI_regression_line <- predict(lm_fit_all, interval="confidence", level = 0.95)

# create prediction interval for the regression line
prediction_interval <- predict(lm_fit_all, interval="predict")
```

```
## Warning in predict.lm(lm_fit_all, interval = "predict"): predictions on current data refer to _futur
```

```r
# plot both confidence interval and the prediction interval
plot(price_bought ~ mileage_bought, data = used_corollas_all,
     main = "Price vs. Mileage for Used Corollas", xlab = "Mileage Bought",
     ylab = "Price Bought")


# plot confidence interval
points(used_corollas_all$mileage_bought, CI_regression_line[, 1], col = "red", type = "l")
points(used_corollas_all$mileage_bought, CI_regression_line[, 2], col = "green", type = "l")
points(used_corollas_all$mileage_bought, CI_regression_line[, 3], col = "green", type = "l")


# plot prediction interval
points(used_corollas_all$mileage_bought, prediction_interval[, 2], col = "blue", type = "l")
points(used_corollas_all$mileage_bought, prediction_interval[, 3], col = "blue", type = "l")
```

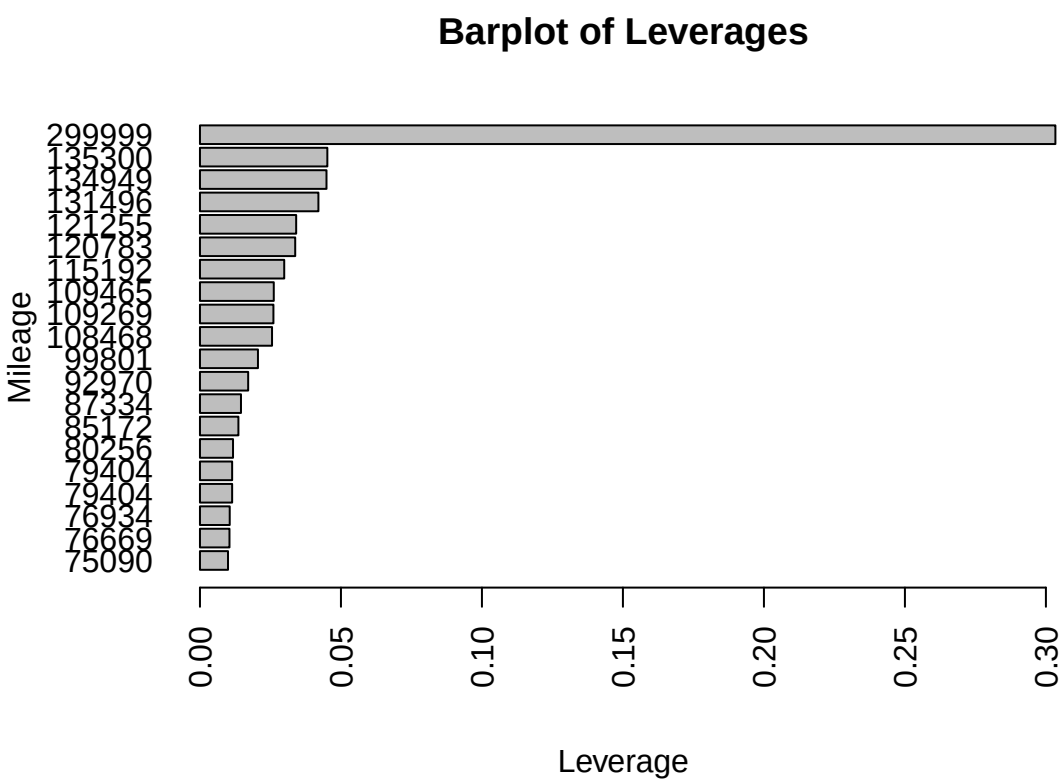


Answers:

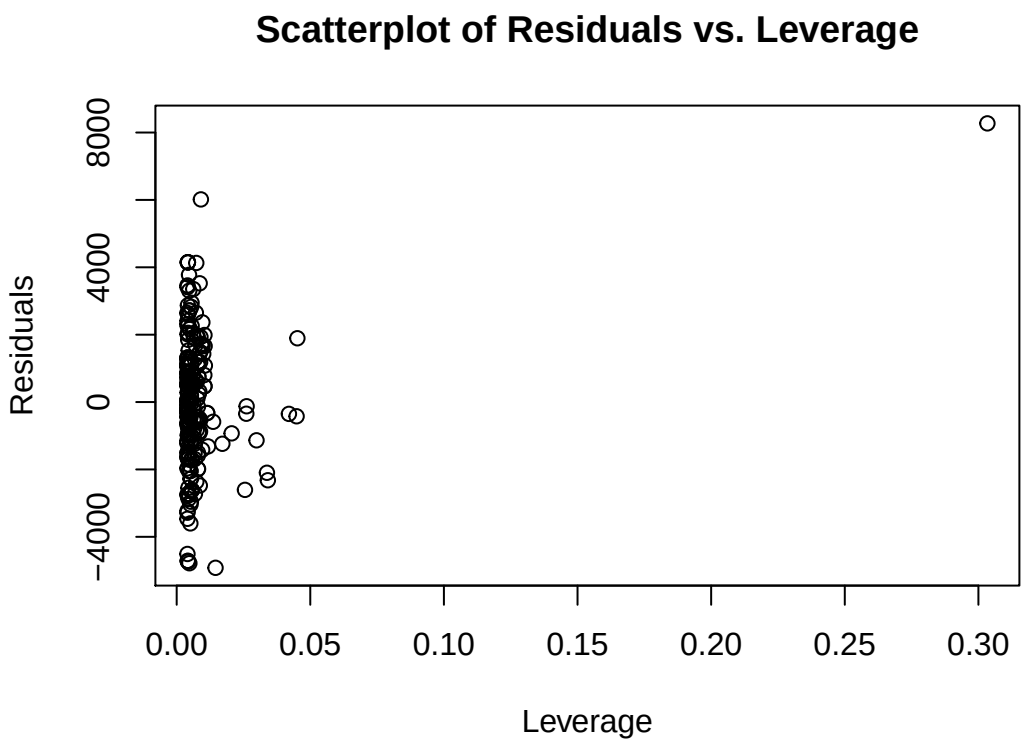
The equation for the confidence interval for the regression line is $\hat{y} \pm t^* \cdot SE_{\hat{\mu}}$, where $SE_{\hat{\mu}} = \sigma_{\in}\sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$. The interval becomes wider for cars that have been driven the most miles because there is more uncertainty at the ends of the line. If we look at the equation we see that as x^*, our x-value, increases or decreases (getting further away from the mean $\bar{x}$), the standard error also increases as the numerator of the SE term gets larger, and therefore the confidence interval also gets wider.

Part 1.5 (10 points): Let's analyze the leverage and Cook's distance for the data points in the `used_corollas_all`. Calculate the leverage for the data points in this model (i.e., the hat values) and plot the 20 largest leverage values found using a bar plot. Also plot the residuals as a function of the leverage for each point, and use R's built in plot functions to plot Cook's distance for each data point and the standardized residuals as a function of the leverage for each point. Based on the 'rules of thumb' we discussed in class, how many points are considered 'very unusual' for the different measures of: Cook's distance, standardized residuals, studentized residuals, and leverage.

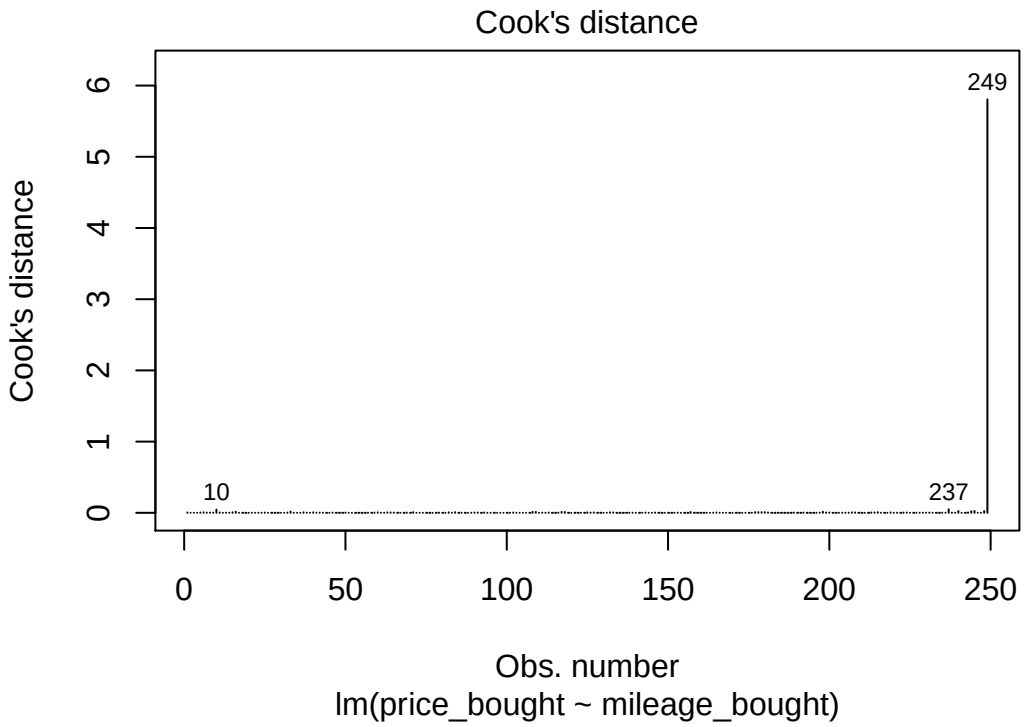
```
# get the h_i "hat values"
hat_vals <- hatvalues(lm_fit_all)
names(hat_vals) <- used_corollas_all$mileage_bought
par(mar = c(6, 6, 4, 4))
barplot(tail(hat_vals, 20), las=2, xlab = "Leverage", horiz = TRUE,
        ylab = "Mileage", mgp=c(4,1,0), main = "Barplot of Leverages")
```



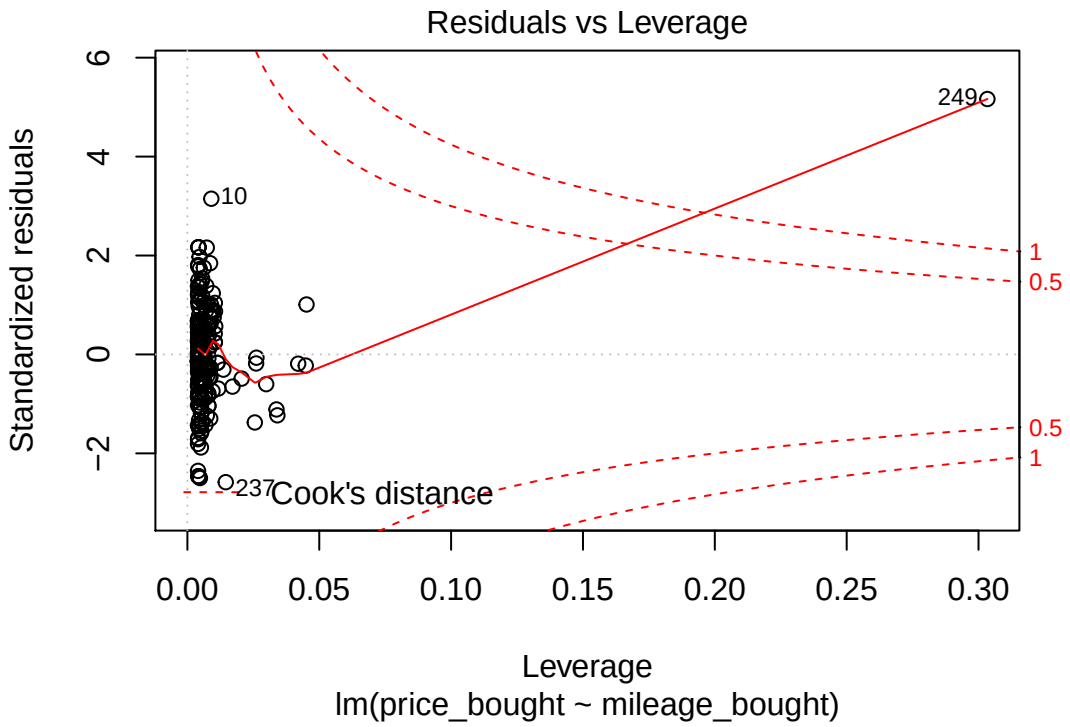

```r
# plot residuals as a function of the hat values
plot(hat_vals, lm_fit_all$residuals,
     xlab = "Leverage",
     ylab = "Residuals",
     main = "Scatterplot of Residuals vs. Leverage")
```




```
# plot Cook's distances and plot them
plot(lm_fit_all, 4)
```




```
# use the base R function to plot the standardized residuals with the hat values
#  along with contours that contain constant Cook's distance values
plot(lm_fit_all, 5)
```




```
# get the number of highly unusual points based on leverage, standardized residuals and studentized res

#Cook's distance:
sum(cooks.distance(lm_fit_all) > 1)
```

```
## [1] 1
```

```
#Leverage: above 3(k+1)/n where k = 1, so above 6/n
sum(hat_vals > 6/249)
```

```
## [1] 10
```

```
#Standardized Residuals: Beyond +/- 3
sum(rstandard(lm_fit_all) > 3, rstandard(lm_fit_all) < -3)
```

```
## [1] 2
```

```
#Studentized Residuals: Beyond +/- 3

sum(rstudent(lm_fit_all) > 3, rstandard(lm_fit_all) < -3)
```

```
## [1] 2
```

Answer

a) Cook's distance: 1 value where $D_i > 1$
b) Standardized Residuals: 2 values
c) Studentized Residuals: 2 values
d) Leverage: 10 values are very unusual

Part 1.6 (5 points): Above you fit two models: `lm_fit_all` which contained all the used Corollas and `lm_fit_150k` which did not contain the high leverage car with 300,000 miles driven. Describe below which you think is best and why? Also describe any limitation to these models.

Answer I think the `lm_fit_150k` model is better in this case because adding the one high leverage car changes the regression line too drastically. The predictions for 150000 miles are probably more accurate with the `lm_fit_150k` model since this number is close to the rest of our data points (and the 300000 mile car is much further away). The limitation to this model is that since the slope is steeper the x-intercept is going to be decreased, so it will become inaccurate faster as the mileage increases. If we are looking at cars with very high mileages perhaps the one including the 300000 miles-driven car would be better. However, of course `lm_fit_all` would be less accurate when looking at smaller mileages.

Part 2: Multiple linear regression

Let's now explore multiple linear regression where we try to predict a response variable y, based on several explanatory variables x_1, x_2 etc.

Part 2.1 (10 points): Let's start again by using dplyr to derive a new data set from the the original Edmunds `car_transactions` data set. Please create a data set in an object called `car_transactions2` that has the following properties:

1) It contains a new variable called `years_old` which is the difference between the year the car was sold, and the model year of the car.
2) It only contains used cars
3) It only contains the variables: price_bought, mileage_bought, years_old, msrp_bought

As a sanity check, if you have created this data frame correctly it should have 17,134 cases

Also, report what is the maximum and minimum value for the variable `years_old`. Explain why the minimum value of `years_old` makes sense (if the value doesn't make sense, read up about purchasing a new car and figure out what is going on). Finally, report what price the least and most expensive used cars sold for.

```
car_transactions <- car_transactions %>%  mutate(years_old = year_sold - model_year_bought)
car_transactions2 <- car_transactions %>% filter(new_or_used_bought == "U") %>% select(price_bought, mi

dim(car_transactions2)
```

```
## [1] 17134     4
```

```r
#Max and min of car "age"
max(car_transactions2$years_old)
```

```
## [1] 34
```

```r
min(car_transactions2$years_old)
```

```
## [1] -1
```

```r
#most and least expensive cars
max(na.omit(car_transactions2$price_bought))
```

```
## [1] 220000
```

```r
min(na.omit(car_transactions2$price_bought))
```

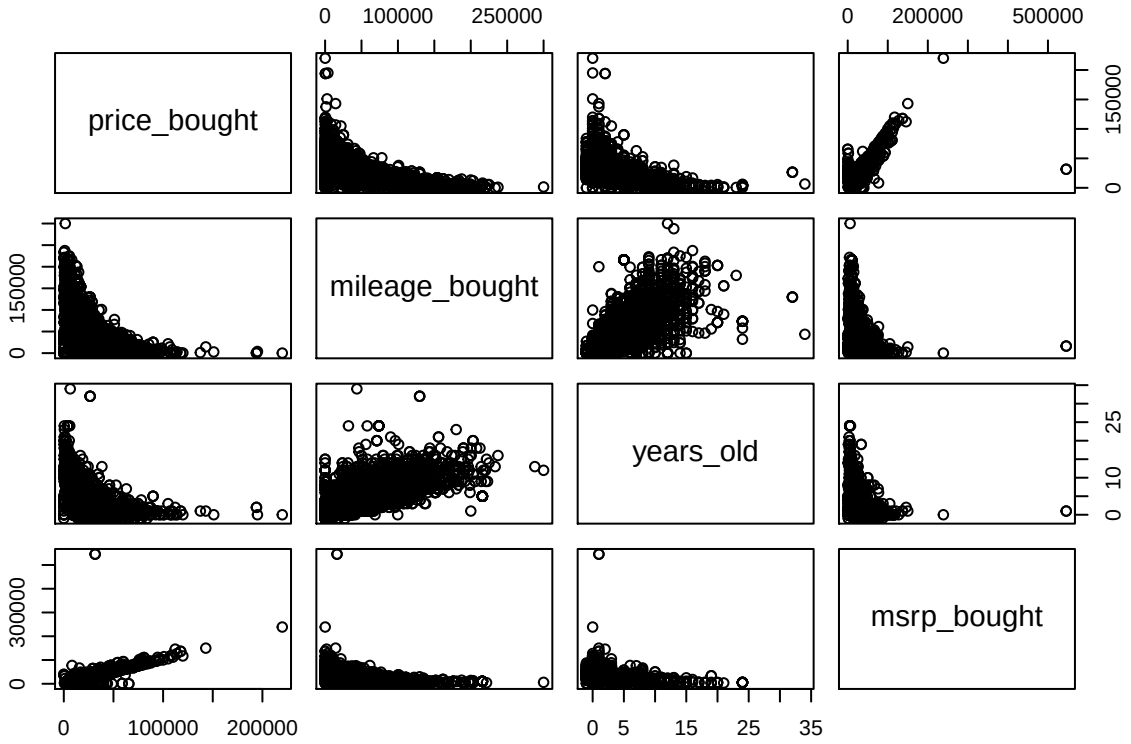
```
## [1] 0
```

Answer:

The maximum value for `years_old` is 34 while the minimum value is -1. This value still makes sense because sometimes car models for the following year are released the year before. For example, if someone could have bought a 2008 model in 2007 but then sold it the same year, making the car "-1" years old. The price of the most expensive car was 220000 and the price of the least expensive one was 0.

Part 2.2 (5 points): Now use the pairs() function to visualize the correlation between all pairs of variables in the `car_transactions2` data frame. Report whether any variable looks like it has a particiularly strong linear relationship with price_bought and whether it makes sense that there would be a strong relationship between these variables.

```r
pairs(car_transactions2)
```



Answer: There seems to be a pretty strong linear relationship between price_bought and msrp_bought. This makes sense because the original value of the car is going to determine the price it is sold for as a used car. A car that is originally a more expensive model will probably seel for more.

Part 2.3 (10 points): Next fit a multiple linear regression model predicting the price a car was bought for using the three variables mileage_bought, years_old, msrp_bought and save the linear fit to an object called `lm_cars`. Then use the summary() function to get information about the the linear regression model you fit by: a) saving the output of the summary() function to an object called `summary_lm_cars`, and b) print the output so you can see the result.

Report below the following information:

a) Do all the regression coefficients for all the variables appear to be statistically significant?
b) Do the signs for the regression coefficients make sense? Explain why.
c) Report what percentage of the total sum of squares is explained by this model by looking at the values stored in the `summary_lm_cars` object.

```
# fit the model
lm_cars <- lm(price_bought ~ mileage_bought + years_old + msrp_bought,
              data = car_transactions2)

# get information about the fit
(summary_lm_cars <- summary(lm_cars))
```

```
##
```

```
## Call:
## lm(formula = price_bought ~ mileage_bought + years_old + msrp_bought,
##     data = car_transactions2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -310279   -2934    -571    2238   63537
##
## Coefficients:
##                    Estimate   Std. Error t value            Pr(>|t|)
## (Intercept)    11155.428797   202.919846   54.98 <0.0000000000000002 ***
## mileage_bought    -0.046924     0.003563  -13.17 <0.0000000000000002 ***
## years_old       -447.222407    40.271434  -11.11 <0.0000000000000002 ***
## msrp_bought        0.608395     0.004918  123.71 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6759 on 8731 degrees of freedom
##   (8399 observations deleted due to missingness)
## Multiple R-squared:  0.7309, Adjusted R-squared:  0.7308
## F-statistic:  7904 on 3 and 8731 DF,  p-value: < 0.00000000000000022
```

```
# look at the percentage of variability explained
summary_lm_cars$r.squared
```

```
## [1] 0.730882
```

Answers:

a) The p-values for all of the regression coefficients are very small (close to 0), so it seems that they are all statistically significant.

b) The signs do all make sense. mileage_bough and year_old have negative signs, which makes sense because we expect the value of the car to decrease as it increases in mileage and gets older. On the other hand, msrp_bought should be positively correlated because the more the car is worth originally, the more it should be worth used.

c) The percentage of total sum squares explained by the model is 73.09% (the unadjusted R-squared value).

Part 2.4 (5 points): Now try to create a model that can account for as much of the variability in the y values by:

a) Using dplyr's mutate function to add new to variables that are derived from the variables in the `car_transactions2` data frame (i.e., square the variables, take logs, add interactions, etc). Save the new data frame to an object called `car_transactions3`.

b) Fit a linear model to the variables in this `car_transactions3`.

c) Calculate the R^2 value

d) Repeat this process to try to generate a R^2 value that is as large as possible, and report what this value is. Whoever get the largest R^2 value in the class will get a prize.

Then use LaTeX to write out the equation for the model you found and report whether you believe this is a good predictive model.

```
car_transactions3 <- car_transactions2 %>%
  mutate(x1 = log10(msrp_bought),
    x2 = years_old*x1,
    x3= mileage_bought*x1,
    x4= msrp_bought*years_old,
    x5 = msrp_bought*mileage_bought,
    x6 = msrp_bought^2,
    x7 = mileage_bought^2,
    x8 = years_old^2,
    x9 = (x6+ x7)^2,
    x10 = (x6 + x8)^2,
    x11 = x1*msrp_bought,
    x12 = x1^2,
    x13 = x12*years_old,
    x14 = x12*msrp_bought,
    x15 = x12*mileage_bought,
    x16 = x7*years_old,
    x17 = x7*msrp_bought,
    x18 = x7*mileage_bought)



lm_cars_3 <- lm(price_bought ~  mileage_bought + years_old + msrp_bought
                + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 +
                  x11 + x12 + x13 +x14 +x15 + x16 + x17 + x18,
              data = car_transactions3)
summary(lm_cars_3)$r.squared
```

```
## [1] 0.9399118
```

```
summary(lm_cars_3)
```

```
##
## Call:
## lm(formula = price_bought ~ mileage_bought + years_old + msrp_bought +
##     x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 + x10 + x11 +
##     x12 + x13 + x14 + x15 + x16 + x17 + x18, data = car_transactions3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -44130  -1199    408   1602  27614
##
## Coefficients:
##                                   Estimate                  Std. Error
## (Intercept)    36298.418858623648702632636  2793.384579129906796879368
## mileage_bought    -1.656434271061017815541     0.181208096325387119085
```

```
## years_old      12782.194818159157875925303  1973.497454876552637870191
## msrp_bought     -155.102589349007018881821     8.960368929844099383786
## x1             18761.408659970911685377359  1269.701031223760082866647
## x2             -8421.369826823631228762679  1146.007343947455638044630
## x3                 0.445320361549493393127     0.108146097714992436845
## x4                -0.052158460999634941035     0.006778981127018952543
## x5                -0.000002871624610158920     0.000000710629260459196
## x6                 0.000025700593907413755     0.000001092797054444631
## x7                 0.000009938551611873941     0.000000283846638999070
## x8                 8.424957671312649054585     4.068653806679009221625
## x9                -0.000000000000000002597     0.000000000000000003008
## x10               -0.000000000000000024902     0.000000000000000003156
## x11               65.946310578620597198096     3.595839765828924949886
## x12            -3434.843276243590935337124   519.516110343833588558482
## x13             1311.621900396983619430102   168.672844391891146642593
## x14               -7.070266373064141518512     0.368601594572947888206
## x15               -0.009784282252063963500     0.016594284687419708774
## x16                0.000000013678142147541     0.000000002877972392517
## x17               -0.000000000058746379396     0.000000000005904905046
## x18               -0.000000000000795716588     0.000000000001451052561
##                t value             Pr(>|t|)
## (Intercept)     12.994 < 0.0000000000000002 ***
## mileage_bought  -9.141 < 0.0000000000000002 ***
## years_old        6.477  0.00000000009867742 ***
## msrp_bought    -17.310 < 0.0000000000000002 ***
## x1              14.776 < 0.0000000000000002 ***
## x2              -7.348  0.00000000000021862 ***
## x3               4.118  0.00003860947806059 ***
## x4              -7.694  0.00000000000001580 ***
## x5              -4.041  0.00005369032298564 ***
## x6              23.518 < 0.0000000000000002 ***
## x7               3.307             0.000948 ***
## x8               2.071             0.038416 *
## x9              -0.863             0.388008
## x10             -7.890  0.00000000000000338 ***
## x11             18.340 < 0.0000000000000002 ***
## x12             -6.612  0.00000000004024828 ***
## x13              7.776  0.00000000000000833 ***
## x14            -19.181 < 0.0000000000000002 ***
## x15             -0.590             0.555462
## x16              4.753  0.00000203931568374 ***
## x17             -9.949 < 0.0000000000000002 ***
## x18             -0.548             0.583451
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3197 on 8713 degrees of freedom
##   (8399 observations deleted due to missingness)
## Multiple R-squared:  0.9399, Adjusted R-squared:  0.9398
## F-statistic:  6490 on 21 and 8713 DF,  p-value: < 0.00000000000000022
```

Answer

We have $\hat{y}$ as the predicted price. The predictive model is $\hat{y} = 32987.45 + -1.823(mileage) + 12782.19(years) + -155.10(msrp) + 18761.40x_1 + -8421.36x_2 + 0.445x_3 + -0.0521x_4 + -0.00000287x_5 + 0.0000257x_6 +$

$0.000000938x_7 + 8.424x_8 + -0.000000000000000002x_9 + -0.000000000000000002x_{10} + 65.94x_{11} + -3434.84x_{12} + 1311.62x_{13} + -7.070x_{14} + -0.00978x_{15} + 0.00000001x_{16} + -0.0000000000587x_{17} + -0.000000000000795x_{18}$ (a long equation!). $x_1, \ldots x_{18}$ are defined in the above code. The R-squared value is 0.9399 which shows the model is a good predictor; however, there are certainly too many variables and there is probably an overfitting problem here, so it is probably not the best model.

Part 2.5 (2 points): If your model in part 2.4 is very large (i.e., has many variables) see if you can come up with a smaller model that captures a reasonable amount of the variability and describe whether you think this new model is better.

```
lm_cars_4 <- lm(price_bought ~  mileage_bought + years_old + msrp_bought
                + x1 + x6 ,
              data = car_transactions3)
summary(lm_cars_4)$r.squared
```

```
## [1] 0.9214954
```

Answer: Here we have far less variables but only a marginally smaller R-squared value, 0.9215. I would say this model is better since we don't has as much as an overfitting problem.

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 8

Homework 9

The purpose of this homework is to learn more about multiple regression models and data wrangling. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday November 17th.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Part 1: Polynomial regression

In the first set of exercises you will get practice running polynomial regression using the IPEDS faculty salary data.

Part 1.1 (2 points): To start, use dplyr to create a data frame called `IPED_2` that only includes schools that have endowments greater that 0 dollars. Also, add a variable to this data frame called `log_endowment` which is the log10 of each school's endowment.

```
load('IPED_salaries_2016.rda')
IPED_2 <- IPED_salaries %>% filter(endowment > 0) %>% mutate(log_endowment = log10(endowment))
```

Part 1.2 (5 points): Fitting polynomial models

Now use polynomial regression to build models that predict total faculty salaires (salary_tot) from log_endowment. Do the polynomial fit for models up to degree 5, and for every model be sure to include the lower order terms as well; i.e., the model of degree 3 should be $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \hat{\beta}_3 x^3$. Save all these models in a list called `poly_models`. Then use the summary() function to print the model of degree 5. Report which coefficients appear to be statistically significant from the degree 5 model.

```
poly_models <- list()
for (i in 1:5){
  poly_models[[i]] <- lm(salary_tot ~ poly(log_endowment, degree = i), data = IPED_2)
}
summary(poly_models[[5]])
```

```
##
## Call:
## lm(formula = salary_tot ~ poly(log_endowment, degree = i), data = IPED_2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -97488 -12157  -2362   9270 167196
##
## Coefficients:
##                                  Estimate Std. Error t value
## (Intercept)                       64565.3      258.1 250.196
## poly(log_endowment, degree = i)1 1067211.9    20672.8  51.624
## poly(log_endowment, degree = i)2  437640.8    20693.8  21.148
## poly(log_endowment, degree = i)3  180102.7    20706.2   8.698
## poly(log_endowment, degree = i)4  -68788.4    20726.7  -3.319
## poly(log_endowment, degree = i)5  -39140.9    20691.5  -1.892
##                                            Pr(>|t|)
## (Intercept)                       < 0.0000000000000002 ***
## poly(log_endowment, degree = i)1 < 0.0000000000000002 ***
## poly(log_endowment, degree = i)2 < 0.0000000000000002 ***
## poly(log_endowment, degree = i)3 < 0.0000000000000002 ***
## poly(log_endowment, degree = i)4             0.000909 ***
## poly(log_endowment, degree = i)5             0.058584 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20630 on 6386 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.3335, Adjusted R-squared:  0.333
## F-statistic: 639.1 on 5 and 6386 DF,  p-value: < 0.00000000000000022
```

Answer:

All of the degrees seem to be statistically significant except degree 5, which has a p-value of 0.058584.

Part 1.3 (7 points): Plotting polynomial models

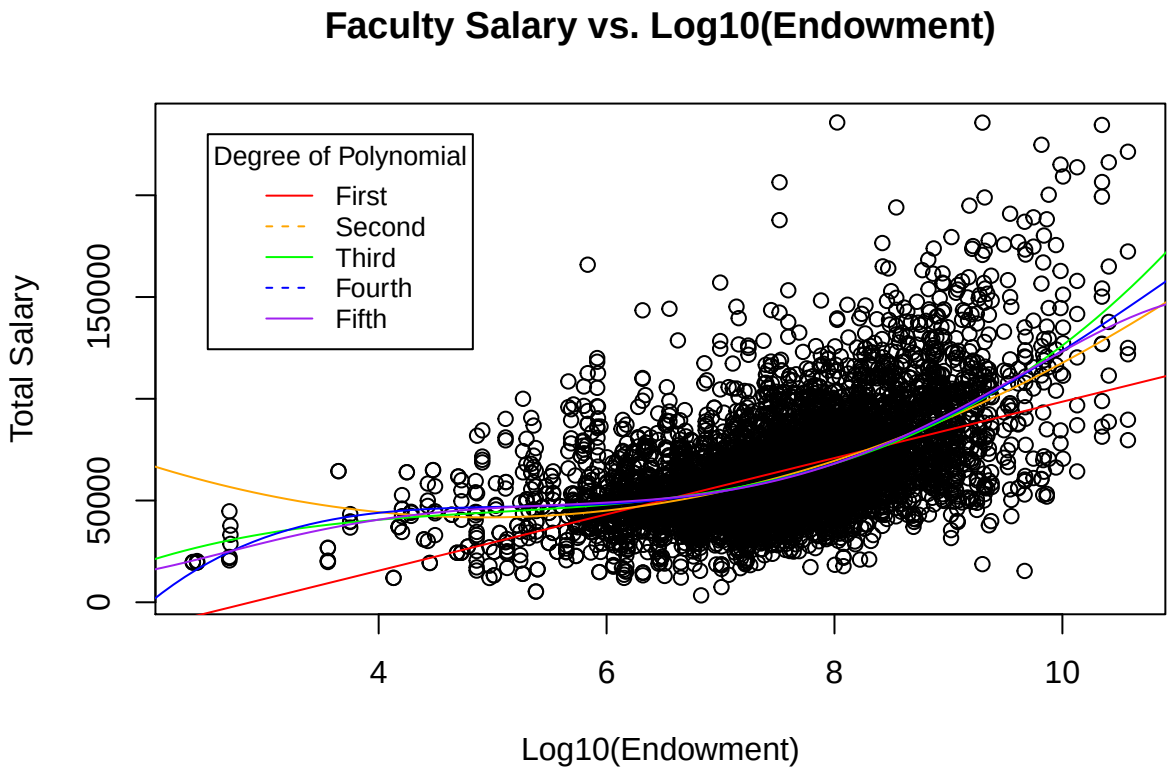
Now visualize these fits of these different polynomial models (i.e., the $\hat{y}$ lines) by creating a scatter plot of the faculty salaries as a function of $log_{10}(endowment)$. Then run a for loop to plot a line for each model fit by:

1) predicting the salaries from a model of the current degree
2) plotting the predicted values as a function of the log_10 endowment in a distinct color (creating a vector with color names outside of the for loop will be helpful).

Try to add a legend to the plot showing what the different colored lines correspond to, and report below which model seems to be the best fit.

```r
# create a data frame for making predictions and a vector of colors
predict_df <- data.frame(log_endowment = seq(0, 13, by = .1))
the_cols <- c("red", "orange", "green", "blue", "purple")

# plot the original data
plot(IPED_2$log_endowment, IPED_2$salary_tot, xlab = "Log10(Endowment)",
     ylab = "Total Salary", main = "Faculty Salary vs. Log10(Endowment)")

# plot each of the model fits
for(i in 1:5){
```

```
  y_vals_predicted <- predict(poly_models[[i]], newdata = predict_df)
  points(predict_df$log_endowment, y_vals_predicted,
     type = "l", col = the_cols[i])
}

#Added a legend
legend(2.5, 230000, legend=c("First", "Second", "Third",
                             "Fourth", "Fifth"),
       col=c("red", "orange", "green", "blue", "purple"),
       title = "Degree of Polynomial", lty=1:2, cex=0.8)
```



Answer
From the plot above, it seems that the fifth degree polynomial is still the best fit.

Part 1.4 (5 points): Extracting R^2 and adjusted R^2 statistics

Now extract the R^2 and adjusted R^2_{adj} statistics. Which model has the largest the R^2 and the largest adjusted R^2_{adj} statistics? Is this what you would expect.

```
all_r_squared <- c()
all_r_adj <- c()
for (i in 1:5){
  all_r_squared[i] <- summary(poly_models[[i]])$r.squared
```

```
  all_r_adj[i] <- summary(poly_models[[i]])$adj.r.squared

}
all_r_squared
```

```
## [1] 0.2774779 0.3240210 0.3319748 0.3331177 0.3334912
```

```
all_r_adj
```

```
## [1] 0.2773648 0.3238094 0.3316611 0.3327000 0.3329693
```

Answer:
The fifth degree polynomial has both the largest R^2 and R^2_{adj} statistics. I expected the R^2 to be the largest, but I thought the addition of another degree would decrease the adjusted value.

Part 1.5 (3 points): Do these models seem reasonable?

Describe overall whether you feel fitting polynomial models here seems like a reasonable thing to do; i.e., pro and cons of using a polynomial model here. There is not necessarily a right answer, just express your thoughts.

Answer
I think fitting a polynomial here does make sense because in reality, all salaries will be within a certain range (above 0 and below some number). This means we shouldn't just use a linear model to predict but should find a model which will account for the flattening of values near the ends of regression. However, there is a disadvantage in that using a polynomial will force the regression into some shape which may not necessarily reflect the nature of salary vs. endowment.

Part 2: Exploring categorical predictors and interactions

Let's now examine how much faculty salaries increase as a function of log endowment size taking into account the rank that different professors have.

Part 2.1 (2 points): Wrangling the data

Start this analysis by creating a data set called `IPED_3` which is modified `IPED_2` in the following way:

1) Only include data from institutions with a CARNEGIE classification of 15 or 31 (these correspond to R1 institutions and liberal arts colleges).

2) Only use the faculty ranks of Lecturer, Assistant, Associate, and Full professors

If you do this right you should `IPED_3` should have 808 rows.

```
IPED_3 <- IPED_2 %>% filter(CARNEGIE %in% c(15,31),
         rank_name %in% c("Lecturer", "Assistant", "Associate", "Full"))
dim(IPED_3)
```

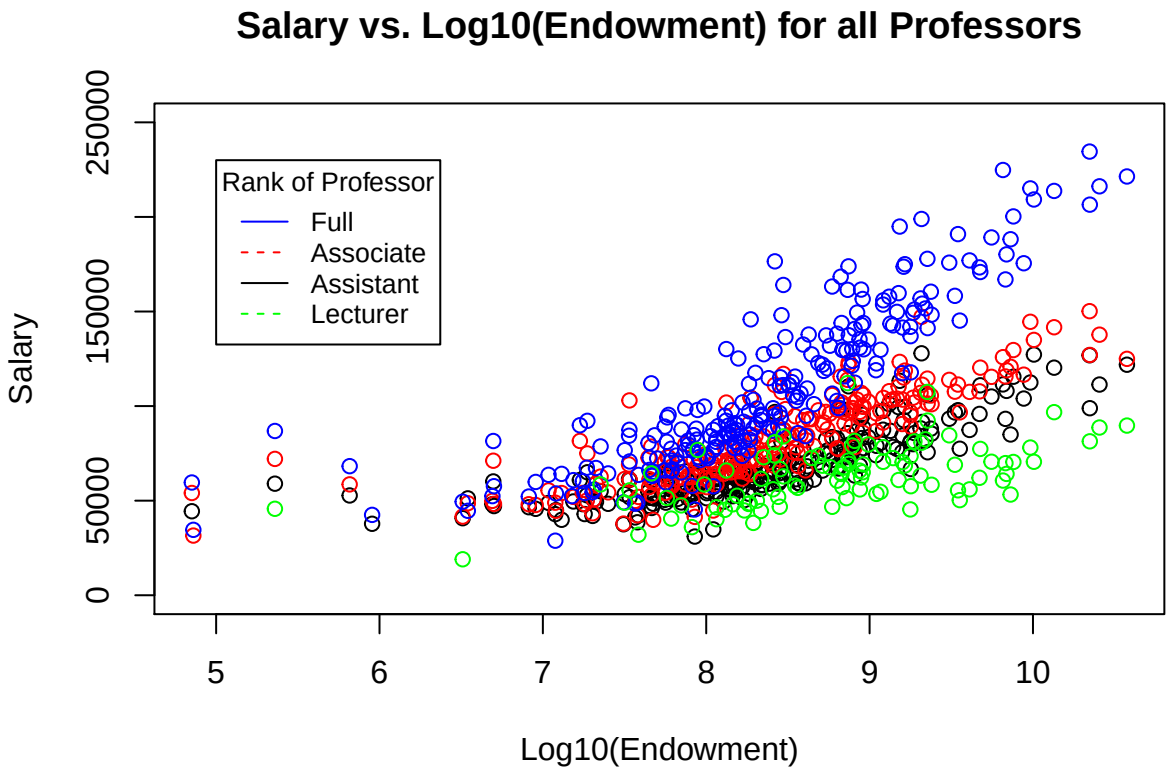
```
## [1] 808  15
```

Part 2.2 (3 points): Visualizing the data

Now create a scatter plot of the data showing the total salary that faculty get paid (salary_tot) as a function of the log endowment size, where each faculty rank is in a different color. In particular, use the following color scheme:

a) Assistant professors are in black
b) Associate professors are in red
c) Full professors are in blue
d) Lecturers are in green

```
plot(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Assistant"),
     ylim = c(0, 250000), ylab = "Salary", xlab = "Log10(Endowment)",
     main = "Salary vs. Log10(Endowment) for all Professors")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Associate"), col = "red")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Full"), col = "blue")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Lecturer"), col = "green")
legend(5, 230000, legend=c("Full", "Associate", "Assistant",
                           "Lecturer"),
       col=c("blue", "red", "black", "green"),
       title = "Rank of Professor", lty=1:2, cex=0.8)
```



Part 2.3 (7 points): Fitting a linear model to the data

Now fit a linear regression model for total salary as a function of log endowment size, but use a separate y-intercept for each of the 4 faculty ranks (and use the same slope for all ranks).

Use the summary() function to extract information about the model, and then answer the following questions about the model:

1) How much do faculty salaries increase for each order of magnitude increase in endowment size?
2) What is the reference faculty rank that the other ranks are being compared to?
3) What is the difference in faculty salaries for each of the other ranks relative to the reference rank?
4) Do there appear to be statistically significant differences between the y-intercept of reference rank and each of the other ranks?
5) How much of the total sum of squares of faculty salary is log10 endowment and faculty rank accounting for in this model based on the R^2 and adjusted R^2 statistics?

```
(intercept_fit <- lm(salary_tot ~ log_endowment + rank_name, data = IPED_3))
```

```
##
## Call:
## lm(formula = salary_tot ~ log_endowment + rank_name, data = IPED_3)
##
## Coefficients:
##        (Intercept)       log_endowment  rank_nameAssociate
##             -106023               25797              -27824
## rank_nameAssistant   rank_nameLecturer
##              -40934              -57334
```

```
summary(intercept_fit)
```

```
##
## Call:
## lm(formula = salary_tot ~ log_endowment + rank_name, data = IPED_3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -53203 -11161  -2275   7280  77600
##
## Coefficients:
##                      Estimate Std. Error t value            Pr(>|t|)
## (Intercept)        -106022.8     6359.8  -16.67 <0.0000000000000002 ***
## log_endowment        25796.8      748.6   34.46 <0.0000000000000002 ***
## rank_nameAssociate  -27823.8     1685.2  -16.51 <0.0000000000000002 ***
## rank_nameAssistant  -40933.9     1685.2  -24.29 <0.0000000000000002 ***
## rank_nameLecturer   -57334.4     2236.6  -25.64 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 18360 on 803 degrees of freedom
## Multiple R-squared:  0.7053, Adjusted R-squared:  0.7038
## F-statistic: 480.5 on 4 and 803 DF,  p-value: < 0.00000000000000022
```

```
(the_coefs_salary <- coef(intercept_fit))
```

```
##        (Intercept)      log_endowment rank_nameAssociate
##          -106022.78           25796.82          -27823.77
## rank_nameAssistant  rank_nameLecturer
##           -40933.90          -57334.39
```

Answers

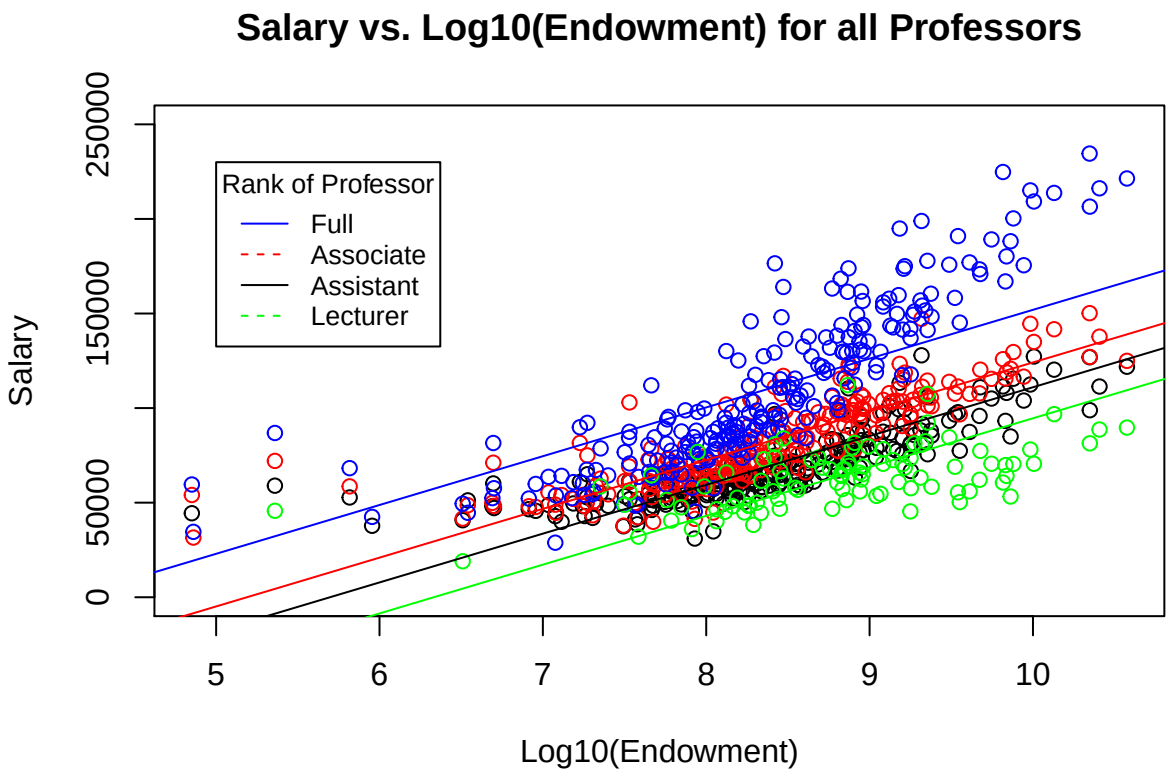
1) The faculty salary increases \$25796.82 for each order of magnitude increase in endowment size.
2) The reference faculty is the Full Professor.
3) Associate Professors are paid \$27923.77 less, Assistant Professors are paid \$40933.90 less, and Lecturers are paid \$57334.39 less.
4) The p-values are all very close to 0 (about $2 * 10^{-16}$), so it appears there is a statistically significant difference between the y-intercept of each rank versus the Full Professor.
5) $R^2 = 0.7053$ and adjusted $R^2 = 0.7038$. So the model accounts for 70.38% of the sum of squares (using the adjusted R-squared).

Part 2.4 (5 points): Visualizing the model fit

Now recreate the scatter plot you created in part 2.2 using the same colors. Now, however, also add on the regression lines with different y-intercepts that you fit in part 2.4 (using the appropriate colors to match the colors of the points).

Are there any situations in particular where using the same slope for each rank seem like it is doing a poor job fitting the data?

```
plot(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Assistant"),
     ylim = c(0, 250000), ylab = "Salary", xlab = "Log10(Endowment)",
     main = "Salary vs. Log10(Endowment) for all Professors")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Associate"), col = "red")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Full"), col = "blue")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Lecturer"), col = "green")

abline(the_coefs_salary[1], the_coefs_salary[2], col = "blue")
abline(the_coefs_salary[1] + the_coefs_salary[3], the_coefs_salary[2], col = "red")
abline(the_coefs_salary[1] + the_coefs_salary[4], the_coefs_salary[2], col = "black")
abline(the_coefs_salary[1] + the_coefs_salary[5], the_coefs_salary[2], col = "green")
legend(5, 230000, legend=c("Full", "Associate", "Assistant",
                           "Lecturer"),
       col=c("blue", "red", "black", "green"),
       title = "Rank of Professor", lty=1:2, cex=0.8)
```



Answer

It seems that using the same slope for a linear regression for Full Professors is poorly fitting the data. This is probably because Full Professors make much higher salaries than other professors and this is likely to increase much more with the amount of money a school has.

Part 2.5 (7 points): Fitting a slightly more complex model

Now fit a linear regression model for total salary as a function of log endowment size, but use separate y-intercepts **and slopes** for each of the 4 faculty ranks. Then answer the following questions:

1) How much of the total sum of squares of faculty salary is this model capturing?
2) Based on this model, if an Associate professor and Full professor both worked at a University that had an endowment of a million dollars, who would get paid more and by how much? Does this seem realistic?

```
(interaction_fit <- lm(salary_tot ~ log_endowment*rank_name, data = IPED_3))
```

```
##
## Call:
## lm(formula = salary_tot ~ log_endowment * rank_name, data = IPED_3)
##
## Coefficients:
##                       (Intercept)                      log_endowment
```

```
##                          -231986                              40888
##               rank_nameAssociate                rank_nameAssistant
##                           125551                             146880
##                rank_nameLecturer  log_endowment:rank_nameAssociate
##                           200710                             -18369
## log_endowment:rank_nameAssistant   log_endowment:rank_nameLecturer
##                           -22482                             -30100
```

```
summary(interaction_fit)
```

```
##
## Call:
## lm(formula = salary_tot ~ log_endowment * rank_name, data = IPED_3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -46914  -9581  -2180   6387  99678
##
## Coefficients:
##                                  Estimate Std. Error t value
## (Intercept)                      -231986       9778 -23.726
## log_endowment                      40888       1165  35.099
## rank_nameAssociate                125551      13987   8.976
## rank_nameAssistant                146881      14124  10.400
## rank_nameLecturer                 200710      20166   9.953
## log_endowment:rank_nameAssociate  -18369       1665 -11.033
## log_endowment:rank_nameAssistant  -22482       1681 -13.377
## log_endowment:rank_nameLecturer   -30100       2311 -13.025
##                                            Pr(>|t|)
## (Intercept)                      <0.0000000000000002 ***
## log_endowment                    <0.0000000000000002 ***
## rank_nameAssociate               <0.0000000000000002 ***
## rank_nameAssistant               <0.0000000000000002 ***
## rank_nameLecturer                <0.0000000000000002 ***
## log_endowment:rank_nameAssociate <0.0000000000000002 ***
## log_endowment:rank_nameAssistant <0.0000000000000002 ***
## log_endowment:rank_nameLecturer  <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15910 on 800 degrees of freedom
## Multiple R-squared:  0.7795, Adjusted R-squared:  0.7776
## F-statistic:   404 on 7 and 800 DF,  p-value: < 0.00000000000000022
```

```
(the_coefs_interaction <- coef(interaction_fit))
```

```
##                       (Intercept)                     log_endowment
##                         -231985.86                          40888.26
##                rank_nameAssociate                rank_nameAssistant
##                          125550.89                         146880.48
##                 rank_nameLecturer  log_endowment:rank_nameAssociate
##                          200710.11                         -18368.58
## log_endowment:rank_nameAssistant   log_endowment:rank_nameLecturer
##                          -22481.85                         -30100.41
```

```
predict(interaction_fit, (data.frame(log_endowment = 6, rank_name = "Full")))
```

```
##        1
## 13343.68
```

```
predict(interaction_fit, (data.frame(log_endowment = 6, rank_name = "Associate")))
```

```
##        1
## 28683.11
```

Answers

1) The adjusted R-squared value is 0.7776, so the model is capturing 77.76% of the sum of squares.
2) The Full professor would get paid $13343.68 and the Associate Professor would get paid $28683. The Associate professor is getting paid $15339.43 more. This seems very unrealistic because a full professor should be getting paid more; also, both of these salaries are far below what any professor in the U.S. should be making.

Part 2.6 (6 points): Visualizing the model

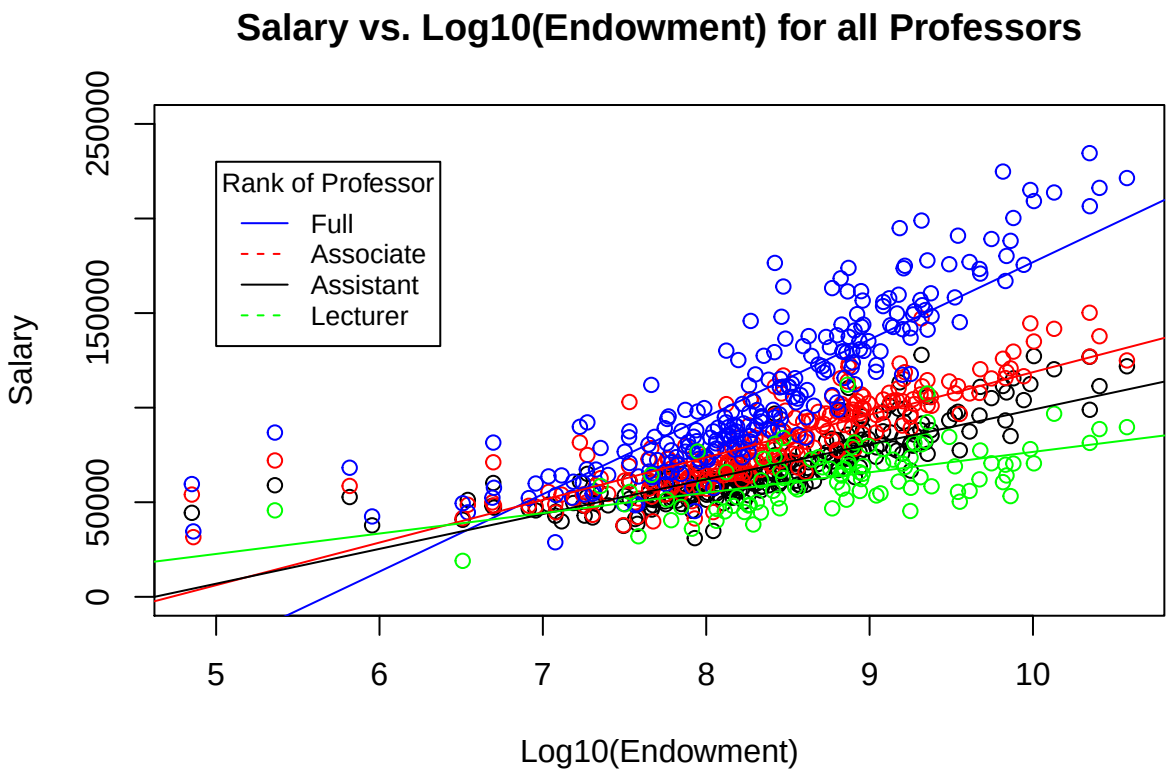
Now again recreate the scatter plot you created in part 2.2 using the same colors. Now, however, also add on the regression line with different y-intercepts and slopes based on the model you fit in part 2.5 (again use the appropriate colors).

Does there seem to be an ordered relationship between ranks and how faculty salary increases with endowment?

```
plot(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Assistant"),
     ylim = c(0, 250000), ylab = "Salary", xlab = "Log10(Endowment)",
     main = "Salary vs. Log10(Endowment) for all Professors")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Associate"), col = "red")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Full"), col = "blue")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Lecturer"), col = "green")

the_coefs_interaction
```

```
##                       (Intercept)                      log_endowment
##                         -231985.86                          40888.26
##                rank_nameAssociate                 rank_nameAssistant
##                          125550.89                          146880.48
##                 rank_nameLecturer log_endowment:rank_nameAssociate
##                          200710.11                          -18368.58
## log_endowment:rank_nameAssistant  log_endowment:rank_nameLecturer
##                          -22481.85                          -30100.41
```

```
abline(the_coefs_interaction[1], the_coefs_interaction[2], col = "blue")
abline(the_coefs_interaction[1] + the_coefs_interaction[3],
       the_coefs_interaction[2] + the_coefs_interaction[6], col = "red")
abline(the_coefs_interaction[1] + the_coefs_interaction[4],
       the_coefs_interaction[2] + the_coefs_interaction[7], col = "black")
abline(the_coefs_interaction[1] + the_coefs_interaction[5],
       the_coefs_interaction[2] + the_coefs_interaction[8], col = "green")
legend(5, 230000, legend=c("Full", "Associate", "Assistant",
                           "Lecturer"),
       col=c("blue", "red", "black", "green"),
       title = "Rank of Professor", lty=1:2, cex=0.8)
```



Answer

There does seem to be an ordered relationship between the rank and the increase in salary. When the endowment is over about 10 million dollars, the Full Professor salary increases the fastest, followed by Associate, Assistant, and then Lecturer. This follows the ranking of actual Professors - the higher the rank, the faster salary will increase.

Part 2.7 (3 points): Comparing models

The model you fit in Part 2.5 is nested within the model you fit in Part 2.3. Use an ANOVA to compare these models. Does adding the additional slopes for each rank seem to improve the model fit?

```
anova(intercept_fit, interaction_fit)
```

```
## Analysis of Variance Table
##
## Model 1: salary_tot ~ log_endowment + rank_name
## Model 2: salary_tot ~ log_endowment * rank_name
##   Res.Df          RSS Df   Sum of Sq      F                Pr(>F)
## 1    803 270762445978
## 2    800 202593228230  3 68169217748 89.729 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer Since the Residual Sum of Squares decreased with the second model, it seems that addition of additional slopes for each rank did improve the model fit. We also see that the p-value for the F-statistic is very small, showing there is a difference in the models' ability to explain the variance.

Part 2.8 (5 points): Improving the model

Can you think of any other ways to improve the model fit? Do an additional analysis where you adjust something about the model or the data to create a better model. Describe below what you did below and why.

```
poly_list <- c()
for (i in 1:5){
poly_list[[i]] <- lm(salary_tot ~ poly(log_endowment, degree = i)*rank_name, data = IPED_3)
}

(poly_fit_5 <- poly_list[[5]])
```

```
##
## Call:
## lm(formula = salary_tot ~ poly(log_endowment, degree = i) * rank_name,
##     data = IPED_3)
##
## Coefficients:
##                                         (Intercept)
##                                              112307
##                   poly(log_endowment, degree = i)1
##                                             1053277
##                   poly(log_endowment, degree = i)2
##                                              374555
##                   poly(log_endowment, degree = i)3
##                                              -69858
##                   poly(log_endowment, degree = i)4
##                                              -94644
##                   poly(log_endowment, degree = i)5
##                                               38416
##                                 rank_nameAssociate
##                                              -29025
##                                 rank_nameAssistant
##                                              -42309
```

```
##                                     rank_nameLecturer
##                                                -53699
## poly(log_endowment, degree = i)1:rank_nameAssociate
##                                               -486877
## poly(log_endowment, degree = i)2:rank_nameAssociate
##                                               -194300
## poly(log_endowment, degree = i)3:rank_nameAssociate
##                                                 3122
## poly(log_endowment, degree = i)4:rank_nameAssociate
##                                                24440
## poly(log_endowment, degree = i)5:rank_nameAssociate
##                                                 5863
## poly(log_endowment, degree = i)1:rank_nameAssistant
##                                               -596970
## poly(log_endowment, degree = i)2:rank_nameAssistant
##                                               -197550
## poly(log_endowment, degree = i)3:rank_nameAssistant
##                                                40694
## poly(log_endowment, degree = i)4:rank_nameAssistant
##                                                43646
## poly(log_endowment, degree = i)5:rank_nameAssistant
##                                               -29350
##  poly(log_endowment, degree = i)1:rank_nameLecturer
##                                               -761827
##  poly(log_endowment, degree = i)2:rank_nameLecturer
##                                               -325137
##  poly(log_endowment, degree = i)3:rank_nameLecturer
##                                                -28598
##  poly(log_endowment, degree = i)4:rank_nameLecturer
##                                                150916
##  poly(log_endowment, degree = i)5:rank_nameLecturer
##                                                 10677
```

```
(summary(poly_fit_5))$adj.r.squared
```

```
## [1] 0.8547149
```

Answer

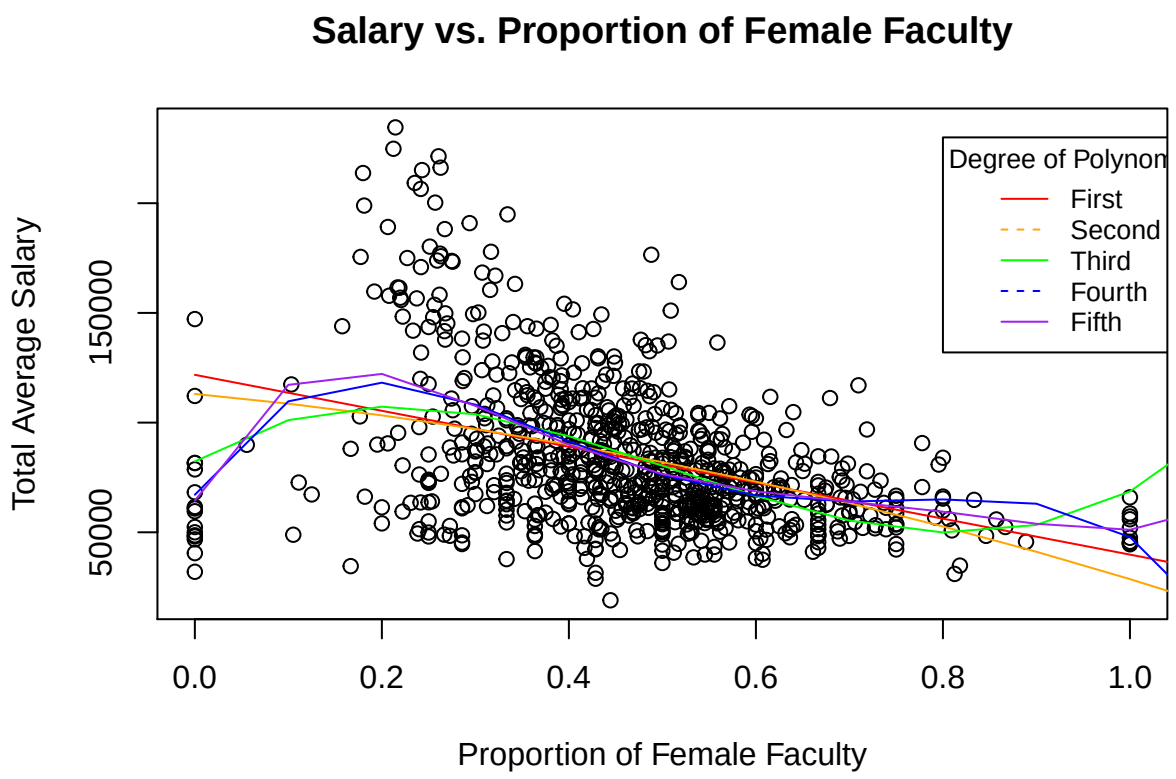
We can use a polynomial regression to try and improve the model fit. We can see from the scatterplot above that there is some curvature to the data, so I thought fitting a polynomial regression might help to address this problem. When I fitted a polynomial regression with degree 5, the R-squared value was 0.8547, so it seems to fit the data better than the linear.

Part 2.9 (5 points): Further explorations

Do an additional exploration or model of the data and report something else interesting.

```
IPED_4 <- IPED_3 %>% mutate(prop_women = num_faculty_women/num_faculty_tot,
                            prop_men = num_faculty_men/num_faculty_tot)
plot(salary_tot ~ prop_women, data = filter(IPED_4),
     ylab = "Total Average Salary", xlab = "Proportion of Female Faculty",
```

```
     main = "Salary vs. Proportion of Female Faculty")
legend(0.8, 230000, legend=c("First", "Second", "Third",
                        "Fourth", "Fifth"),
       col=c("red", "orange", "green", "blue", "purple"),
       title = "Degree of Polynomial", lty=1:2, cex=0.8)

poly_models_4 <- list()
for (i in 1:5){
  poly_models_4[[i]] <- lm(salary_tot ~ poly(prop_women, degree = i), data = IPED_4)
}

predict_df_1 <- data.frame(prop_women = seq(0, 13, by = .1))
for(i in 1:5){

  y_vals_predicted <- predict(poly_models_4[[i]], newdata = predict_df_1)
  points(predict_df_1$prop_women, y_vals_predicted,
     type = "l", col = the_cols[i])
}
```




```
r_squared <- c()
r_adj <- c()
for (i in 1:5){
  r_squared[i] <- summary(poly_models_4[[i]])$r.squared
  r_adj[i] <- summary(poly_models_4[[i]])$adj.r.squared
```

```
}

r_squared
```

```
## [1] 0.1609012 0.1649185 0.2193693 0.2515759 0.2546323
```

```
r_adj
```

```
## [1] 0.1598601 0.1628438 0.2164565 0.2478478 0.2499853
```

```
summary(poly_models_4[[1]])
```

```
##
## Call:
## lm(formula = salary_tot ~ poly(prop_women, degree = i), data = IPED_4)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -89771 -18990  -4469  15511 130351
##
## Coefficients:
##                              Estimate Std. Error t value
## (Intercept)                     84037       1088   77.24
## poly(prop_women, degree = i)  -384495      30928  -12.43
##                                        Pr(>|t|)
## (Intercept)                  <0.0000000000000002 ***
## poly(prop_women, degree = i) <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30930 on 806 degrees of freedom
## Multiple R-squared:  0.1609, Adjusted R-squared:  0.1599
## F-statistic: 154.6 on 1 and 806 DF,  p-value: < 0.00000000000000022
```

Answer I compared the proportion of female faculty members at each university (using IPED_3 which already filtered out the four levels of professors, and the types of colleges). Then I fitted polynomial regressions up to degree five. For the linear regression I found a negative slope of -364495. This seemed to be statistically significant as the p-value was very close to 0, showing that there is some negative relationship between the proportion of female faculty memebers and the average salary at the school. However, none of the polynomial models seemed to fit the data well as all R-squared values were less than 0.3. Looking at the data points I believe there is some correlation though, so I think this may be worth looking more into. Perhaps there really is a wage gap, or perhaps the colleges with more female faculty are all smaller colleges with less money to pay professors.

Part 3: More data wrangling

Thanksgiving is coming up which means a lot of Americans will be traveling. In particular, since New Haven is relatively close to New York City, so it is likely that a number of people will be flying out of airports in

the New York City area for the holiday. A major frustration to flying is when a flight is delayed. Let's use dplyr to do some quick explorations of the data to some ways to potentially avoid flight delays.

Let's start by loading data for flights leaving New York City in 2013. Use *? flights* for more information about the data set. You don't need to modify anything on the code below.

```
#install.packages("nycflights13")
# get the flight delays data and load dplyr
require("nycflights13")
data(flights)
data(airlines)   # the names of the airline carriers
```

Part 3.1 (5 points): Flights that start off with a delay might end up making up some time during the course of the flight. Test whether this is true on average. Hint: only use flights that have positive departure delay.

```
delayed_flights <- flights %>% filter(dep_delay > 0)

mean(delayed_flights$dep_delay - delayed_flights$arr_delay, na.rm = TRUE)
```

```
## [1] 4.607264
```

```
median(delayed_flights$dep_delay - delayed_flights$arr_delay, na.rm = TRUE)
```

```
## [1] 7
```

```
dep_delayed <- delayed_flights$dep_delay
arr_delayed <- delayed_flights$arr_delay

t.test(dep_delayed, arr_delayed)
```

```
##
##  Welch Two Sample t-test
##
## data:  dep_delayed and arr_delayed
## t = 21.291, df = 254640, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  4.306395 5.179653
## sample estimates:
## mean of x mean of y
##  39.37323  34.63021
```

Answers:

The mean difference between departure delays and arrival delays is 4.607, meaning on average, a delayed flight makes up 4.607 minutes during the course of a flight. The median of this value is 7. The t-test shows there is a statistically significant difference between the means so it does seem to be true on average.

Part 3.2 (5 points): One way to avoid being delayed would be to avoid the worst airlines. Which airline had the longest arrival delays on average, and how long was this average delay? Use the *airlines* data frame to figure out which airline each carrier code corresponds to.

```
# get the average delay for each airline

avg_arr_delay <- flights  %>%  group_by(carrier) %>%
   summarize(mean_arr_delay = mean(arr_delay, na.rm = TRUE))
arrange(avg_arr_delay, desc(mean_arr_delay))[1,]
```

```
## # A tibble: 1 x 2
##   carrier mean_arr_delay
##   <chr>            <dbl>
## 1 F9                21.9
```

```
carrier <- toString(avg_arr_delay[which.max(avg_arr_delay$mean_arr_delay),][1])
(airlines[which(airlines$carrier == carrier),])[2]
```

```
## # A tibble: 1 x 1
##   name
##   <chr>
## 1 Frontier Airlines Inc.
```

Answers:
Frontier Airlines had the longest average delay of 21.92 minutes.

Part 3.3 (5 points): Another way to avoid flight delays would be to avoid particularly bad times to fly. Which month of the year had the longest departure delays? Also report which hour of the day had the longest departure delays. Finally, report how many flights left at the hour of the day that had the longest delay and what the average deley was at that time.

```
(month_delay <- flights  %>%  group_by(month) %>%
   summarize(mean_arr_delay = mean(arr_delay, na.rm = TRUE))
 %>% arrange(desc(mean_arr_delay)))[1,]
```

```
## # A tibble: 1 x 2
##   month mean_arr_delay
##   <int>          <dbl>
## 1     7           16.7
```

```
(hour_delay <- flights  %>%  group_by(hour) %>%
   summarize(mean_arr_delay = mean(arr_delay, na.rm = TRUE))
  %>% arrange(desc(mean_arr_delay)))[1,]
```

```
## # A tibble: 1 x 2
##    hour mean_arr_delay
##   <dbl>          <dbl>
## 1    21           18.4
```

```r
dim(filter(flights, hour == 21))
```

```
## [1] 10933    19
```

Answers:

July had the longest departure delay of 16.71 minutes.

The hour with the longest departure delays was 21:00 (9pm) with a time of 18.39 minutes. There were 10933 flights that left at this time.

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 9

Homework 10

The purpose of this homework is to learn about methods selecting models and to practice data visualization. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday November 24th.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Part 1: Model selection

In class 20 we fit polynomial models of up to degree 5 for predicting the price of a used Toyota Corolla from the number of miles driven. We showed that the R^2 value for this fit always increases (or stays the same) as more variables (or a higher degrees polynomials) are added to the model. Thus if one was to judge how "good" a model was based on the R^2 statistic value, one would always choose a very complex model that has likely has a lot of variables.

As we also discussed in class, there are other statistics and methods that can potentially give better measures of how well a model fits the data. In the set of exercises below, you will empirically evaluate these different methods for selecting models to see how well they work.

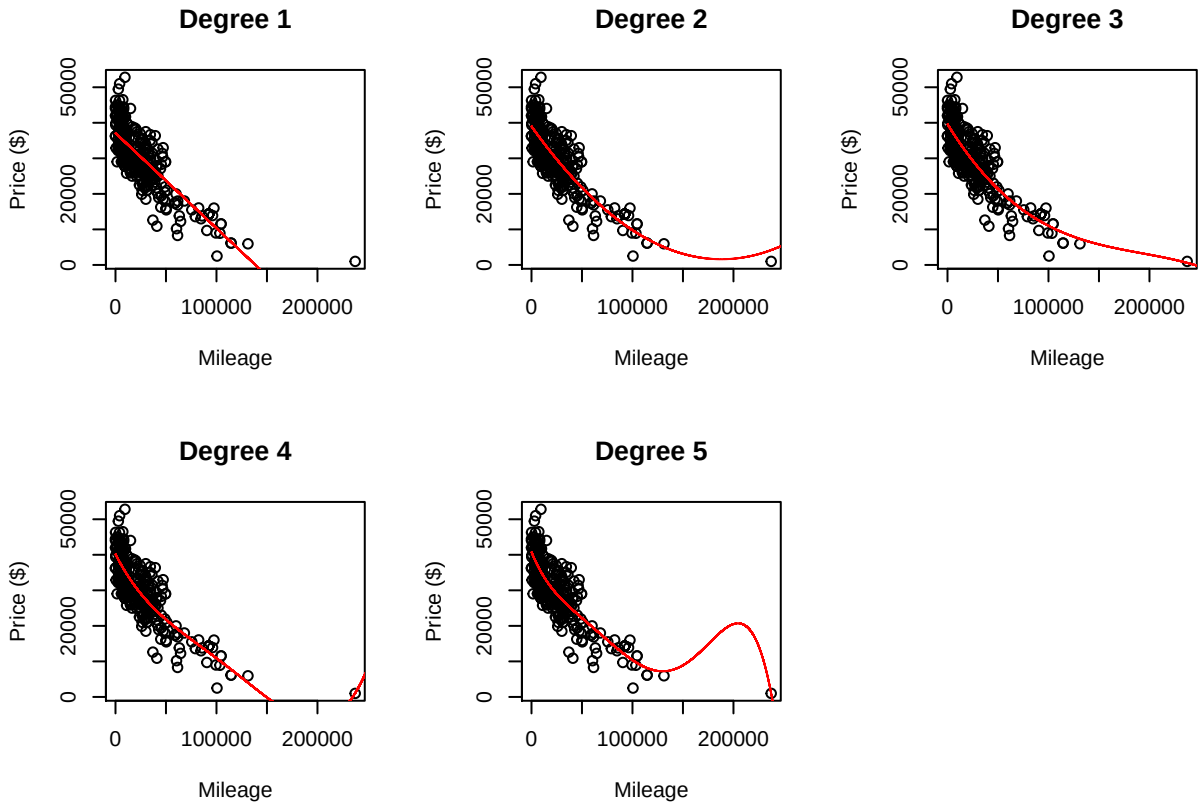
Part 1.1 (3 points):

The code below loads the Edmunds car transaction data and creates a data frame that has data from the used BWM model '3 Series'. It also plots the model fits for predicting price as a function of miles driven for models up to degree 5.

Based on looking at the model fits, which degree polynomial do you think is the best fit to the data?

```
load('car_transactions.rda')
car_model_name <-  "3 Series"   #  "MAZDA3"
used_cars <- select(car_transactions, price_bought,mileage_bought,
                    model_bought, make_bought, new_or_used_bought) %>%
  filter(model_bought == car_model_name, new_or_used_bought == "U") %>%
  na.omit(.)
par(mfrow = c(2, 3))
x_vals_df <- data.frame(mileage_bought = 0:300000)
for (i in 1:5){

  curr_model <- lm(price_bought ~ poly(mileage_bought, degree = i), data = used_cars)

  model_summary <- summary(curr_model)

  y_vals_predicted <- predict(curr_model, newdata = x_vals_df)
```

```r
  plot(price_bought ~ mileage_bought, data = used_cars, xlab = "Mileage",
       ylab = "Price ($)", main = paste("Degree", i))

  points(x_vals_df$mileage_bought, y_vals_predicted, type = "l", col = "red")


}
```



Answers Just looking at the model fits, it seems that the Degree 3 polynomial is the best fit to the data. I believe the degree 3 polynomial fits the points the most accurately since it is continually decreasing (unlike degree 2, 4, and 5 polynomials, although the 2nd degree does seem to be a good fit also) and accounts for the curvature when the mileage increases to very high.

Part 1.2 (10 points): Now let's try calculating different measure of model fit for polynomials from degree 1 to degree 5. Use a for loop to that loops over models of degree 1 to degree 5 and saves the following statistics:

1) R^2: save to a vector called `all_r_squared`
2) R^2_{adj}: save to a vector called `all_r_squared`
3) AIC: save to a vector called `all_aic`
4) BIC: save to a vector called `all_bic`

Then use the which.max() function and which.min() function to determine which model each of these statistics would select. Fill in the table below by placing an x in the appropriate column for the model that each

statistic would select (you will fill in the last line in part 1.3). Also comment on which statistics you think are leading to the best model choice.

```
all_r_squared <- NULL
all_adj_r_squared <- NULL
all_aic <- NULL
all_bic <- NULL
# create a for loop to extract the relevant statistics
# print the degree selected by each model selection method
#using the which.max() or which.min() functions

for (i in 1:5){

  curr_model <- lm(price_bought ~ poly(mileage_bought, degree = i), data = used_cars)

  model_summary <- summary(curr_model)

  all_r_squared[i] <- model_summary$r.squared
  all_adj_r_squared[i] <- model_summary$adj.r.squared
  all_aic[i] <- AIC(curr_model)
  all_bic[i] <- BIC(curr_model)

}


which.max(all_r_squared)
```

```
## [1] 5
```

```
which.max(all_adj_r_squared)
```

```
## [1] 5
```

```
which.min(all_aic)
```

```
## [1] 3
```

```
which.min(all_bic)
```

```
## [1] 2
```

Answer

	1	2	3	4	5
R^2					x
R^2_{adj}					x
AIC			x		
BIC		x			
cross-val		x			

Answer

I think the AIC and BIC models are leading to the best model choice since they are penalizing models with a greater number of predictors (as we can see from the graphs, the degree 5 polynomial is not a very good predictor as the R^2 values may suggest). I think in this case AIC is a bit better than BIC because even thought BIC allows us to choose a model with less predictors, it seems that the third degree polynomial is a bit of a better fit than the second.

Part 1.3 (10 points):

As we also discussed in class, we can use cross-validation to assess model fit, but building a model on a *training set* of data and then evaluating the accuracy of the predictions on a separate *test set* of data. When evaluating the model, we can use the *mean squared prediction error* (MSPE) as a measure of how accurately the model is making predictions on new data. This measure is defined as:

$MSPE = \frac{1}{m}\Sigma_{i=1}^{m}(y_i - \hat{y}_i)^2$ where the y_i come from the m points in the *test set*.

For more information on this measure, see wikipedia.

The code below splits the data in half and creating a *training set* that we will use to learn the estimated coefficients $\hat{\beta}$'s and also creating a *test set* with the other half of the data that we can evaluate how accuerately the model can make predictions. Use a for loop to create models of degree 1 to 5 using the training data, have the models predictions on the test set, and then calculate the MSPE based on their predictions. Add to the table above when model has the minimum MPSE by putting an x in the appropriate column and print out the result below to show your work.

```
# create the training set and the test set
total_num_points <- dim(used_cars)[1]
num_training_points <- floor(total_num_points/2)
training_data <- used_cars[1:num_training_points, ]
test_data <- used_cars[(num_training_points + 1):total_num_points, ]
# run a for loop to calculate the MSPE for models of degree 1 to 5
MSPE_values <- c()
for(i in 1:5){
  fit_cv <- lm(price_bought ~ poly(mileage_bought, degree = i), data = training_data)
  test_predictions_1 <- predict(fit_cv, newdata = test_data)
  MSPE_values[i] <- mean((test_data$price_bought - test_predictions_1)^2)
}
# then find the model with the minimal MSPE
which.min(MSPE_values)
```

```
## [1] 2
```

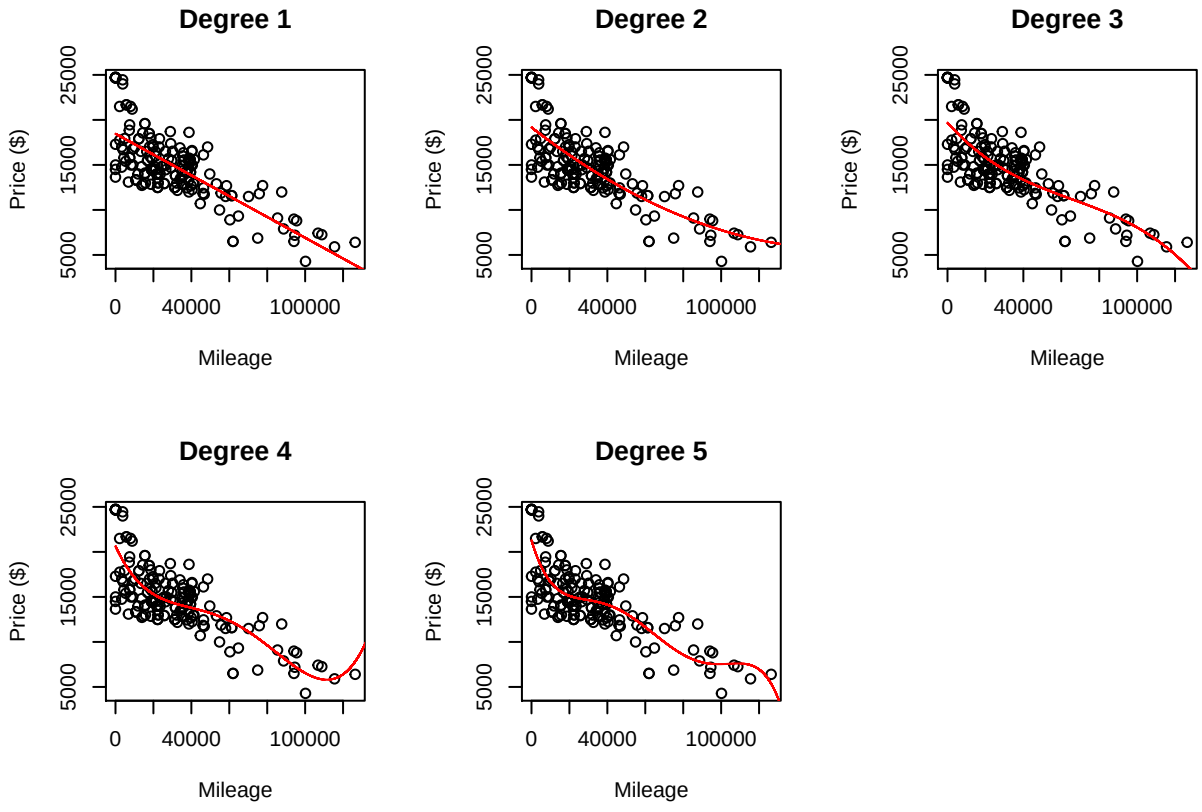
Part 1.4 (10 points):

Now rerun parts 1.1 to 1.3 but using only Mazda 3's instead of BMW 3 series (e.g., filter for "MAZDA3"). If you have written your code in a flexible way you should really only have to change the line "3 Series" to "Mazda 3" in part 1.1 and the rest of the code should run. Then fill out the table below for Mazda 3's, and answer all the following questions for the Mazda 3 data:

a) From looking at the models fits, which degree model seems to fit the data best
b) which statistics you think are leading to the best model choice

c) overall (for both car brands) which model selection method do you think is working best (and why).

```
#Part 1.1
car_model_name <-  "MAZDA3"
used_cars <- select(car_transactions, price_bought,mileage_bought, model_bought, make_bought, new_or_us
  filter(model_bought == car_model_name, new_or_used_bought == "U") %>%
  na.omit(.)
par(mfrow = c(2, 3))
x_vals_df <- data.frame(mileage_bought = 0:300000)
for (i in 1:5){

  curr_model <- lm(price_bought ~ poly(mileage_bought, degree = i), data = used_cars)

  model_summary <- summary(curr_model)

  y_vals_predicted <- predict(curr_model, newdata = x_vals_df)

  plot(price_bought ~ mileage_bought, data = used_cars, xlab = "Mileage",
       ylab = "Price ($)", main = paste("Degree", i))

  points(x_vals_df$mileage_bought, y_vals_predicted, type = "l", col = "red")
}
```




```
#Part 1.2
all_r_squared <- NULL
all_adj_r_squared <- NULL
```

```r
all_aic <- NULL
all_bic <- NULL
# create a for loop to extract the relevant statistics
# print the degree selected by each model selection method
#using the which.max() or which.min() functions

for (i in 1:5){

  curr_model <- lm(price_bought ~ poly(mileage_bought, degree = i), data = used_cars)

  model_summary <- summary(curr_model)

  all_r_squared[i] <- model_summary$r.squared
  all_adj_r_squared[i] <- model_summary$adj.r.squared
  all_aic[i] <- AIC(curr_model)
  all_bic[i] <- BIC(curr_model)

}


which.max(all_r_squared)
```

```
## [1] 5
```

```r
which.max(all_adj_r_squared)
```

```
## [1] 5
```

```r
which.min(all_aic)
```

```
## [1] 5
```

```r
which.min(all_bic)
```

```
## [1] 5
```

```r
#Part 1.3
# create the training set and the test set
total_num_points <- dim(used_cars)[1]
num_training_points <- floor(total_num_points/2)
training_data <- used_cars[1:num_training_points, ]
test_data <- used_cars[(num_training_points + 1):total_num_points, ]
# run a for loop to calculate the MSPE for models of degree 1 to 5
MSPE_values <- c()
for(i in 1:5){
  fit_cv <- lm(price_bought ~ poly(mileage_bought, degree = i), data = training_data)
  test_predictions_1 <- predict(fit_cv, newdata = test_data)
  MSPE_values[i] <- mean((test_data$price_bought - test_predictions_1)^2)
}
# then find the model with the minimal MSPE
which.min(MSPE_values)
```

```
## [1] 2
```

Answer

	1	2	3	4	5
R^2					X
R^2_{adj}					X
AIC					X
BIC					X
cross-val		X			

a) From looking at the plots it seems that the degree 2 polynomial since it most closely resembles how we would expect car prices to decline.

b) All of the statistics except the Mean Squared Prediction Error point towards the same model (the 5th degree polynomial), which doesn't seem to be the best model choice. Therefore, MSPE seems to be the best statistic for choosing a model. Among the first four statistics we considered at first, I think AIC and BIC would tend to lead to the best model (although they did not) since they are penalizing for the number of predictors we have, whereas R^2 will always increase. Adjusted R^2 also accounts for the number of predictors but the values usually end up very similar to R^2. If we are looking for a smaller model, BIC is more effective than AIC.

c) It seems that cross-validation worked best across both models of cars since it was the only method which chose correctly for the Mazda 3's, and chose one of the two possible correct fits for BMWs. This makes sense as cross validation is testing whether the model actually works on the data, whereas other methods simply use all of the data to make a predictive model.

Bonus question (0 points):

Run three-fold cross-validation where split the data into 3 parts and you:

a) learn the model parameter estimates on two of the data splits (2/3rds of the data)
b) make predictions on the last data split (1/3 of the data)
c) repeat steps *a* and *b* leaving a different test set out each time and training on the other 2 splits
d) average the MSPE results over the 3 cross-validation splits.
e) repeat this procedure for models of degree 1-5 (i.e., by having nested for loops)

Does the three-fold cross-validation results lead to the same conclusions as using only 1 training and test split?

```
# create the training set and the test set
total_num_points <- dim(used_cars)[1]
training_points_1 <- floor(total_num_points/3)
training_points_2 <- floor(2*total_num_points/3)
training_data_1 <- used_cars[1:training_points_1, ]
training_data_2 <- used_cars[(training_points_1+1):training_points_2, ]
training_data_3 <- used_cars[(training_points_2+1):total_num_points, ]

#First 2
# run a for loop to calculate the MSPE for models of degree 1 to 5
```

```r
MSPE_values <- c()
training_1 <- rbind(training_data_1, training_data_2)
training_2 <- rbind(training_data_1, training_data_3)
training_3 <- rbind(training_data_2, training_data_3)


for(i in 1:5){
  fit_cv_1 <- lm(price_bought ~ poly(mileage_bought, degree = i), data = training_1)
  test_predictions_1 <- predict(fit_cv_1, newdata = training_data_3)
  MSPE_val_1 <- mean((training_data_3$price_bought - test_predictions_1)^2)

  fit_cv_2 <- lm(price_bought ~ poly(mileage_bought, degree = i), data = training_2)
  test_predictions_2 <- predict(fit_cv_2, newdata = training_data_2)
  MSPE_val_2 <- mean((training_data_2$price_bought - test_predictions_2)^2)

  fit_cv_3 <- lm(price_bought ~ poly(mileage_bought, degree = i), data = training_3)
  test_predictions_3 <- predict(fit_cv_3, newdata = training_data_1)
  MSPE_val_3 <- mean((training_data_1$price_bought - test_predictions_3)^2)

  MSPE_values[i] <- mean(MSPE_val_1, MSPE_val_2, MSPE_val_3)
}
# then find the model with the minimal MSPE
which.min(MSPE_values)
```

```
## [1] 2
```

Answer

Training on two thirds of the data set still gives us the second degree polynomial as the best model.

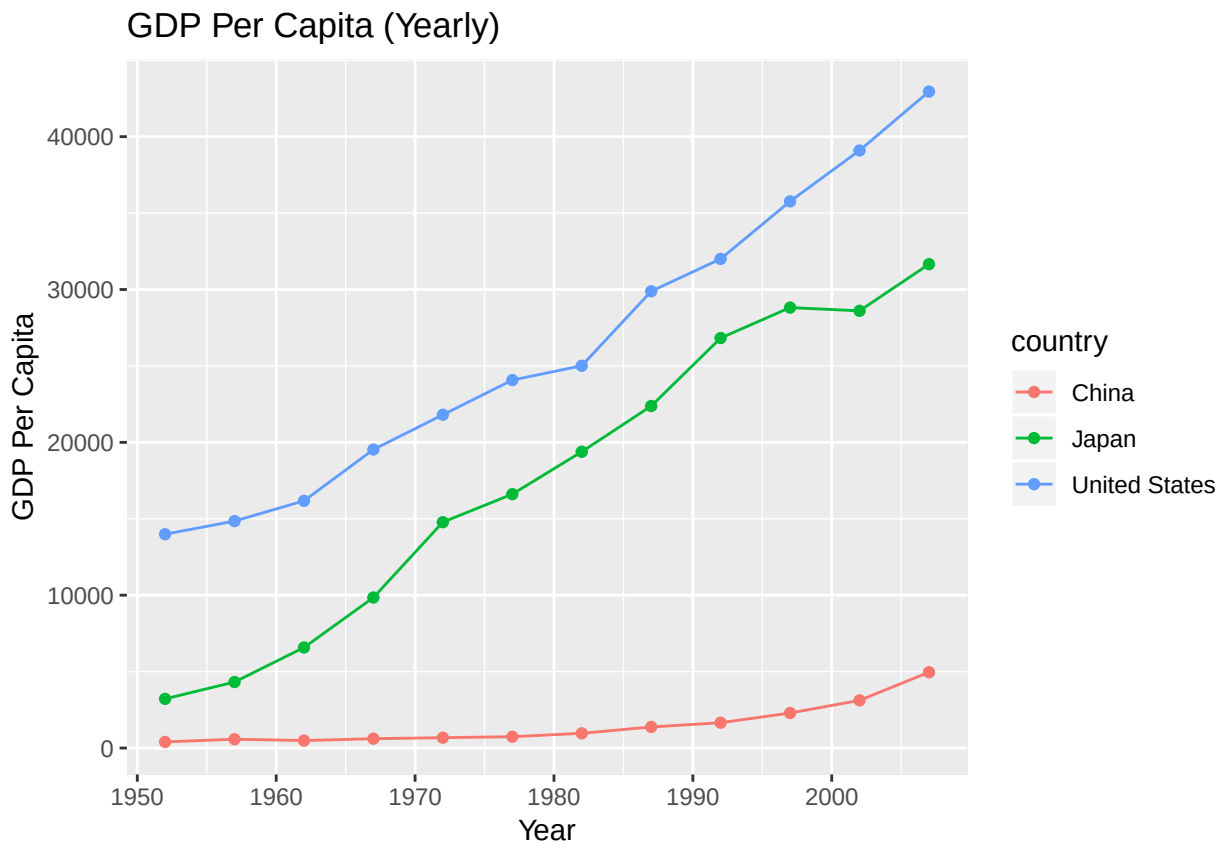
Part 2: Data visualization

In the next set of exercises you will use ggplot2 to compare different visualizations and see which gives the clearest insights. A useful resource for ggplot and other tidyverse code is the book R for Data Science.

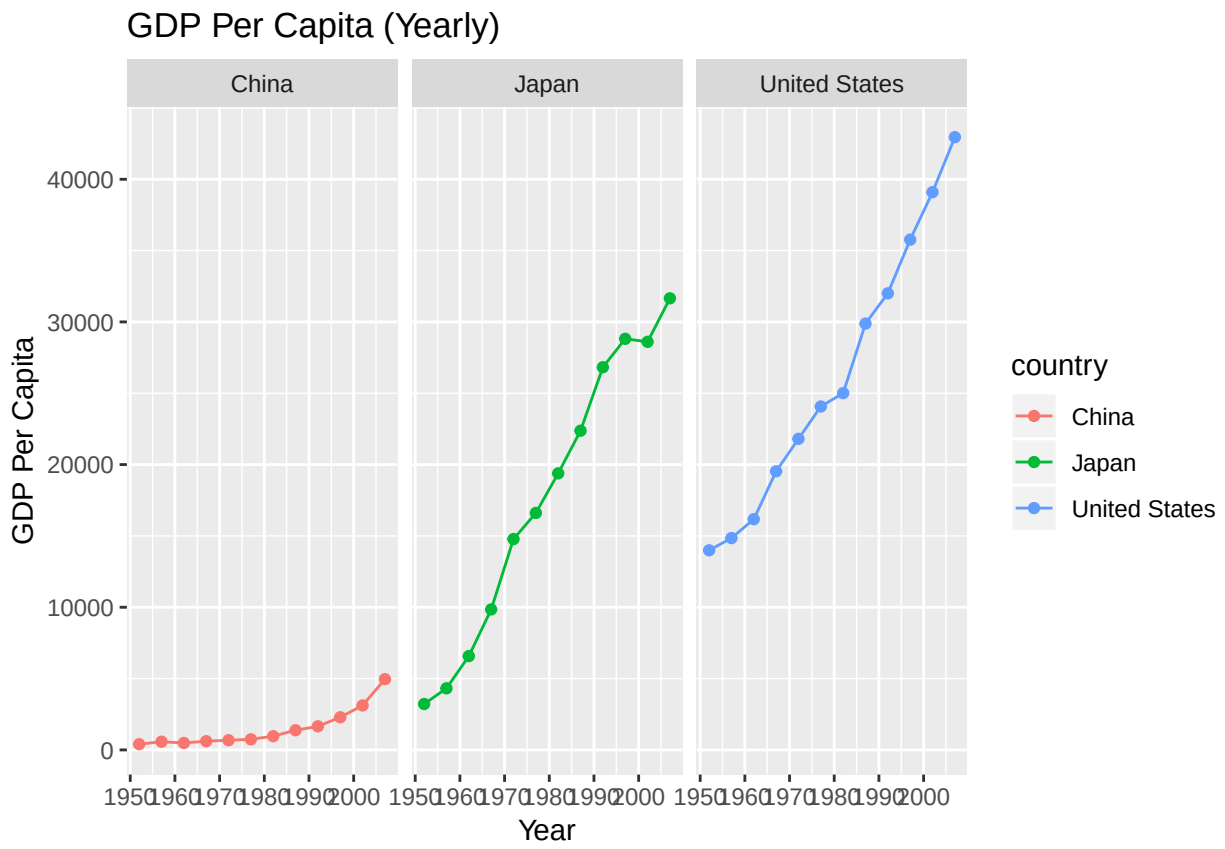
Part 2.1 (10 points): Let's start by comparing some visualizations on the gapminder data which contains information about different countries in the world over time. Use ggplot and the gapminder data to compare the GDP per capita of Japan, the United States and China. Plot a line graph of GDP per capita as a function of the year, with each country in a different color. Also, create a plot that compares these countries GDP per capita as a funciton of the year using facets, where the data from each country is on a separate subplot. As always, make sure to label your axes in this plot and in all other plots in this worksheet. Do you think one type of plots is better than another in comparing these countries? Explain why. (Hint for completing this exercise: first use the dplyr filter() function to get the subset of data you need, then plot it).

```r
# filter the data to get only data from Japan the United States and China
gapminder_filtered <- gapminder %>% filter(country == "Japan" |
                                           country == "United States" |
                                           country == "China")
```

```
# create a line plot showing the three countries on the same plot
ggplot(gapminder_filtered, aes(x = year, y = gdpPercap, col = country)) +
  geom_line() + geom_point() + labs(fill = "Country", x = "Year",
  y = "GDP Per Capita", title = "GDP Per Capita (Yearly)")
```




```
# use facets to put the three countries on side-by-side plots
ggplot(gapminder_filtered, aes(x = year, y = gdpPercap, col = country)) +
  geom_line() + geom_point() + labs(x = "Year",
  y = "GDP Per Capita", title = "GDP Per Capita (Yearly)") +
  facet_wrap(~country)
```



Answers: [Explain whether you think of of these plots is more informative than the other].

I think the second plot with each country on its own subplot is more informative. Since the scales for each plot are the same we can still see how each country's GDP per capita varies in magnitude, as with the first plot. However, the side-by-side plots give us a better sense for how rapidly the GDP of Japan and the United States grew as compared to China, and also show us that China's GDP has begun a steep rise (while still being far below either of the other two countries). Overall, I feel that the two plots are still pretty similar and show the same information.

Part 2.2 (10 points): DataExpo is a Statistics event at the Joint Statistical Meetings where different researchers compare data analysis methods applied to a common data set. In 2018, the data set consisted of weather predictions made between 2014 and 2017. In this exercise, let's look at the data from this event and try to visualize the prediction accuracies for predictions made 0 to 6 days in the future.

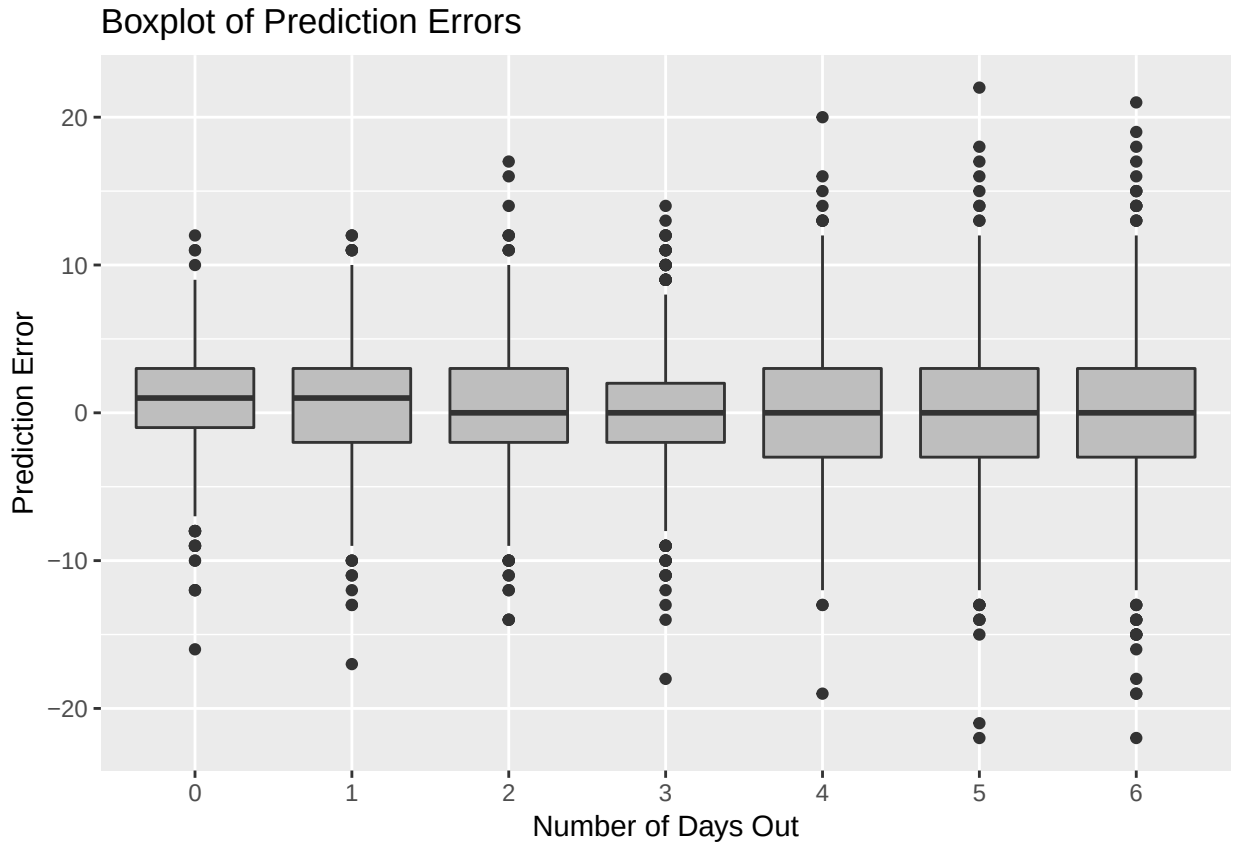
The code below loads a data frame called `forecast_ne_joined` that has the prediction errors for the maximum temperature for the 9 cities in New England, along with several other variables. First, create a new data frame called `new_haven_preds` that has only the predictions from New Haven, and has only the variables cityID, city, num_days_out_prediction_made and max_temp_prediction_error. Also, type convert the variable `num_days_out_prediction_made` to a factor using the mutate() and as.factor() functions. Then use ggplot to create plots that compare the prediction accuracy as a function of the number of days in advance that a prediction was made using the following geoms:

1) Create a boxplot using geom_boxplot()
2) Create a violin using geom_violin()
3) Create a joy plot using geom_density_ridges()

Note that the geom geom_density_ridge() comes from the ggridges packages that was loaded at the top of the worksheet, and that the x and y aesthetic mapping is in the opposite order as the mapping used for the geom_boxplot() and geom_violin() geoms.

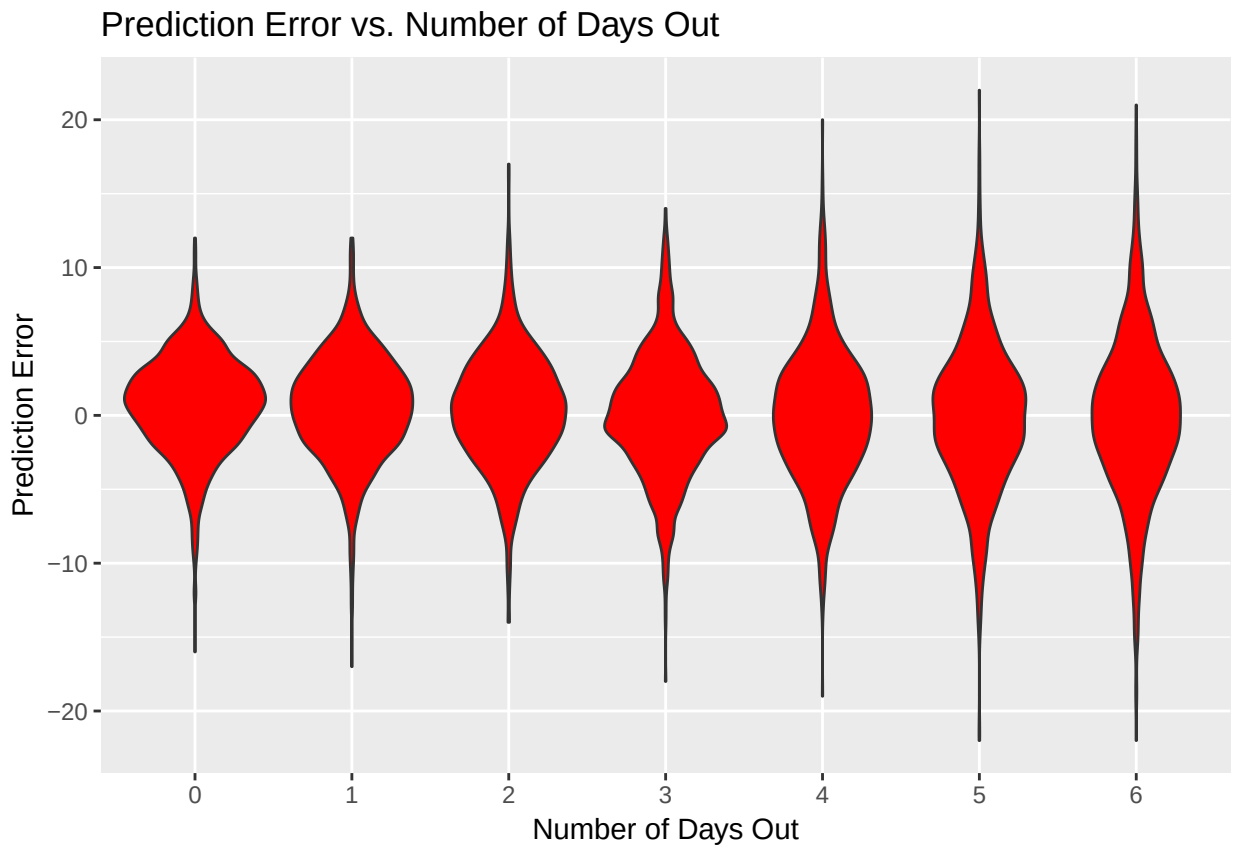
After you created these plots, briefly discuss which plot you believe which most clearly shows how the prediction accuracy decreases as a function of days in the future. Also, don't forget to label your axes using the xlab() and ylab() functions.

```
# load the data that has the weather prediction errors
forecast <- read.csv('forecast_ne_joined.csv')
# filter and select the data to get data from only New Haven for the 4 variables of interest
forecast_filtered <- forecast %>% filter(city == "New Haven") %>%
  select(cityID, city, num_days_out_prediction_made, max_temp_prediction_error)
# and convert num_days_out_prediction_made to a factor using the mutate() function
forecast_filtered <- forecast_filtered %>% mutate(num_days_out_prediction_made =
                                                    as.factor(num_days_out_prediction_made))
# compare the predictions made 0 to 6 days out using box plots
ggplot(forecast_filtered, aes(x = num_days_out_prediction_made,
                              y = max_temp_prediction_error)) +
                              geom_boxplot(fill = "gray") +
            xlab("Number of Days Out") + ylab("Prediction Error") +
            ggtitle("Boxplot of Prediction Errors")
```

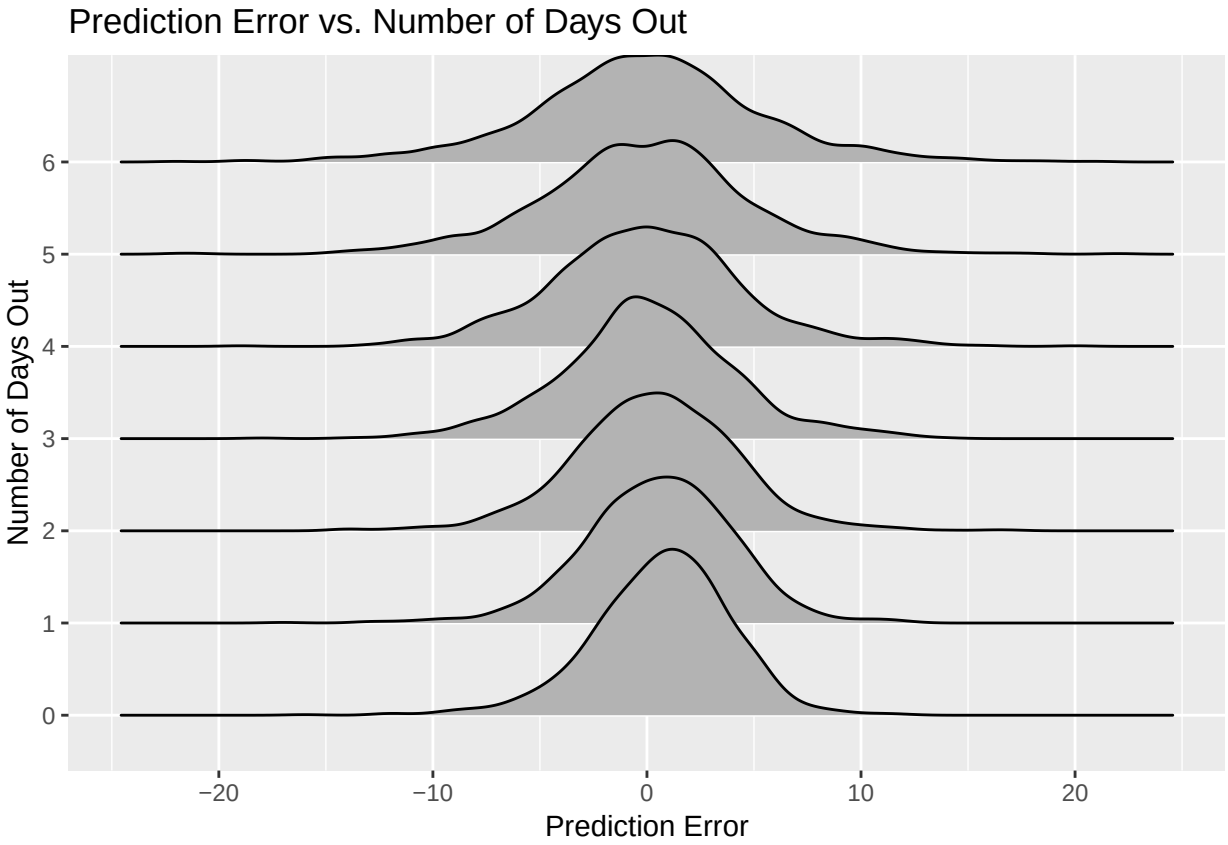



```
# compare the predictions made 0 to 6 days out using violin plots
ggplot(forecast_filtered, aes(x = num_days_out_prediction_made,
                              y = max_temp_prediction_error))+
```

```
  geom_violin(fill = "red") + labs(x = "Number of Days Out",
                                   y = "Prediction Error",
                                   title = "Prediction Error vs. Number of Days Out")
```




```
# compare the predictions made 0 to 6 days out using joy plots
ggplot(forecast_filtered, aes(y = num_days_out_prediction_made,
                              x = max_temp_prediction_error ))+
  geom_density_ridges() + labs(y = "Number of Days Out",
                                   x = "Prediction Error",
                                   title = "Prediction Error vs. Number of Days Out")
```



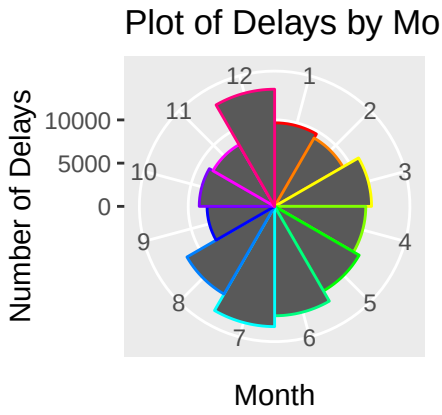
Answers:

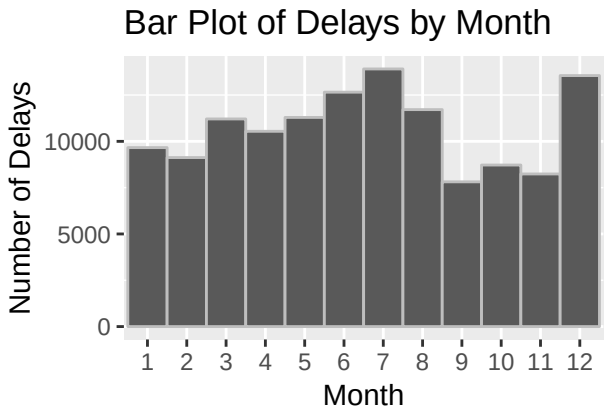
I think the joy plot shows most clearly how the prediction accuracy decreases as a function of days in the future. From this plot we can see that the prediction error is clustered at a higher density around 0 when the Number of Days Out is lower, and the peak becomes shorter and wider as the number of days increases. This visual representation is very clear since we can see the distributions increasing progressively in width. I think this progression can also be seen with the violin and box plots, but is not as clear because we don't have a way of determining whether the points outside the middle 50 percent actually make the entire prediction error increase.

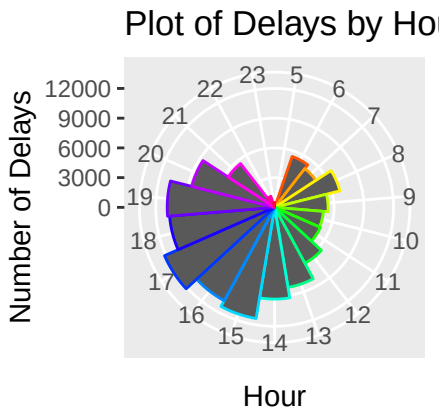
Part 2.3 (10 points): Create an *interesting* plot using one of the data sets we have discussed in class or another data set you can find. Try exploring other features of ggplot we have not discussed in class using the ggplot cheat sheet. See if you can find something interesting in the data and explain why you find it interesting.

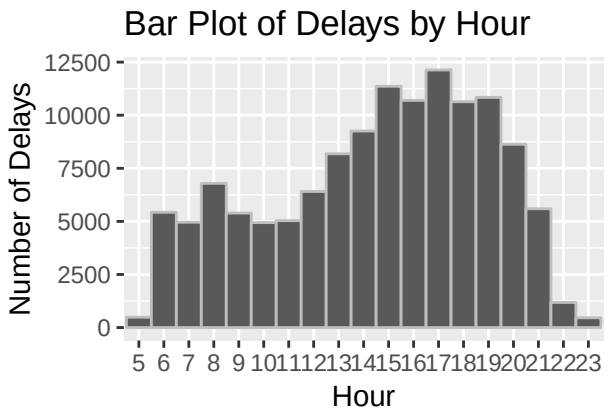
```
require("nycflights13")
data(flights)
data(airlines)

#As in Pset 9, I took the flights data and filtered out delayed flights,
#then grouped by month and hour.
delayed_flights <- flights %>% filter(dep_delay > 0)

month_delay <- flights  %>%  group_by(month) %>%
   summarize(mean_arr_delay = mean(arr_delay, na.rm = TRUE))%>%
```

```r
  arrange(desc(mean_arr_delay))

hour_delay <- flights  %>%  group_by(hour) %>%
   summarize(mean_arr_delay = mean(arr_delay, na.rm = TRUE)) %>%
  arrange(desc(mean_arr_delay))

#Coxcomb Chart of each month and the number of delays
require(gridExtra)
plot1 <- ggplot(delayed_flights, aes(x = factor(month))) +
  geom_bar(width = 1, colour = rainbow(n = 12)) + coord_polar() +
  xlab("Month") + ylab("Number of Delays") + ggtitle("Plot of Delays by Month")
plot2 <- ggplot(delayed_flights, aes(x = factor(month))) +
  geom_bar(width = 1, colour = "gray") +
  xlab("Month") + ylab("Number of Delays") + ggtitle("Bar Plot of Delays by Month")

#Coxcomb chart of each hour and the number of delays
plot3 <- ggplot(delayed_flights, aes(x = factor(hour))) +
  geom_bar(width = 1, colour = rainbow(19)) + coord_polar()+
  xlab("Hour") + ylab("Number of Delays") + ggtitle("Plot of Delays by Hour")

plot4 <- ggplot(delayed_flights, aes(x = factor(hour))) +
  geom_bar(width = 1, colour = "gray") +
  xlab("Hour") + ylab("Number of Delays") + ggtitle("Bar Plot of Delays by Hour")

grid.arrange(plot1, plot2, plot3, plot4, ncol = 2)
```









Answers:

From the last pset we analyzed the flights dataset and found that for months, July had the longest average delay, and in hours, 9pm had the longest delays. However, using ggplot I created two Coxcomb charts showing the nubmer of total delays for the dataset grouped by month and by hour. After visualizing we can see that the highest *number* of delays occurred during the month of December and at the time of 5pm. I think this information is probably better presented in a bar plot which is why I included the barplots side by side with the Coxcomb plot. The Coxcomb plot is probably better for presentation if we are aiming to make more impact on a less knowledgeable audience. Another reservation I have with this visualization is that the chart only shows the number of delays and not the proportion. These numbers then absolutely make sense because more people travel during the summer and December (vacation months), and in the evenings, so we would expect more problems and delays at those times.

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 10